

L Number	Hits	Search Text	DB	Time stamp
-	2	(711/\$.ccls. and 707/\$.ccls.) and (file with (block or partition) same size same (unallocat\$3 or unuse\$3) same (accomodat\$3 or reserv\$3))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/10 19:31
-	1	711/\$.ccls. and 707/\$.ccls. and WAFL	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/10 19:31
-	6	file with (block or partition) same size same (unallocat\$3 or unuse\$3) same (accomodat\$3 or reserv\$3)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/10 19:32
-	6	file and (block or partition) and (direct and indirect) and size and (unallocat\$3 or unuse\$3) and (accomodat\$3 or reserv\$3) and exceed\$3 and write and subtract\$3 and flag and 711/\$.ccls. and 707/\$.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/10 19:32
-	2	711/\$.ccls. and 707/\$.ccls. and ("zero length" same write)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/10 19:33
-	8	711/\$.ccls. and 707/\$.ccls. and ("zero length")	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/10 19:34
-	1	(711/\$.ccls. and 707/\$.ccls. and 709/\$.ccls.) and (711/\$.ccls. and 707/\$.ccls. and ("zero length"))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/10 19:38
-	69	(file adj server)and (file adj size) and (direct or indirect) same block\$1 and exceed\$3	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/10 19:39
-	58	(file adj server)and (file adj size) and (direct or indirect) same block\$1 and exceed\$3 and flag	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/10 20:00
-	266	file and (block or partition) and (direct and indirect) and size and (unallocat\$3 or unuse\$3) and (accomodat\$3 or reserv\$3) and exceed\$3 and write and subtract\$3 and flag	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/10 20:01
-	6	file and (block or partition) and (direct and indirect) and size and (unallocat\$3 or unuse\$3) and (accomodat\$3 or reserv\$3) and exceed\$3 and write and subtract\$3 and flag and 711/\$.ccls. and 707/\$.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/10 20:01
-	4	709/\$.ccls. and WAFL	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/10 20:02
-	51	(file adj server)and (file adj size) and (direct or indirect) same block\$1 and exceed\$3 and flag and cache\$2	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/10 20:04
-	50	(file adj server)and (file adj size) and (direct or indirect) same block\$1 and exceed\$3 and flag and cache\$2 and write!	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/06/10 20:04

-	423	file and (block or partition) and (direct and indirect) and size and (unallocat\$3 or unuse\$3) and (accomodat\$3 or reserv\$3) and exceed\$3 and write and subtract\$3	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM TDB	2003/06/10 20:05
-	240	(reserv\$3 or accomodat\$3 or sav\$3) with (unlocat\$3 or unused) same block\$2	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM TDB	2003/06/11 21:02
-	15	file with (block or partition) same(reserv\$3 or accomodat\$3 or sav\$3) with (unlocat\$3 or unused) same block\$2	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM TDB	2003/06/11 21:01
-	272	Hitz.in.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM TDB	2003/06/11 21:02
-	0	Hitz.in. and ((reserv\$3 or accomodat\$3 or sav\$3) with (unlocat\$3 or unused) same block\$2)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM TDB	2003/06/11 21:03
-	9	Hitz.in. and ((reserv\$3 or accomodat\$3 or sav\$3) and (unlocat\$3 or unused) same block\$2)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM TDB	2003/06/11 21:27
-	10	Lewis.in. and ((reserv\$3 or accomodat\$3 or sav\$3) and (unlocat\$3 or unused) same block\$2)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM TDB	2003/06/11 21:31
-	1	(Blake and Lewis).in. and ((reserv\$3 or accomodat\$3 or sav\$3) and (unlocat\$3 or unused) same block\$2)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM TDB	2003/06/11 21:34

Set	Items	Description
S1	15	WRITE()ANYWHERE()FILE()SYSTEM()LAYOUT OR WALF
File-	2:INSPEC	1969-2003/Jun W1
		(c) 2003 Institution of Electrical Engineers
File	34:SciSearch(R)	Cited Ref Sci 1990-2003/Jun W2
		(c) 2003 Inst for Sci Info
File	94:JICST-EPlus	1985-2003/Jun W2
		(c)2003 Japan Science and Tech Corp(JST)
File	95:TEME-Technology & Management	1989-2003/May W4
		(c) 2003 FIZ TECHNIK
File	144:Pascal	1973-2003/May W4
		(c) 2003 INIST/CNRS
File	275:Gale Group Computer DB(TM)	1983-2003/Jun 13
		(c) 2003 The Gale Group

ETC

Set	Items	Description
S1	50	AU='LEWIS B' OR AU='LEWIS BLAKE'
S2	54	AU='PATEL K' OR AU='PATEL KAYURI'
S3	23	AU='CHEN RAY':AU='CHEN RAYMOND H'
S4	125	S1 OR S2 OR S3
S5	25	S4 AND IC=G06F?

File 347:JAPIO Oct 1976-2003/Feb(Updated 030603)  
(c) 2003 JPO & JAPIO

File 348:EUROPEAN PATENTS 1978-2003/Jun W01  
(c) 2003 European Patent Office

File 349:PCT FULLTEXT 1979-2002/UB=20030605,UT=20030529  
(c) 2003 WIPO/Univentio

File 350:Derwent WPIX 1963-2003/UD,UM &UP=200337  
(c) 2003 Thomson Derwent

5/5/1 (Item 1 from file: 348)  
DIALOG(R) File 348:EUROPEAN PATENTS  
(c) 2003 European Patent Office. All rts. reserv.

01469862

**INSTANT SNAPSHOT**

**UNMITTELBARER SCHNAPPSCHUSS**

**INSTANTANE**

**PATENT ASSIGNEE:**

Network Appliance, Inc., (2617422), 495 East Java Drive, Sunnyvale,  
California 94089, (US), (Applicant designated States: all)

**INVENTOR:**

**LEWIS, Blake** , 2316 Greer Road, Palo Alto, CA 94303, (US)  
**EDWARDS, John, K.**, 1173 Crandano Court, Sunnyvale, CA 94087-2076, (US)  
**VISWANATHAN, Srinivasan**, 751 Saltillo Place, Fremont, CA 94536, (US)

**LEGAL REPRESENTATIVE:**

Leeming, John Gerard (74731), J.A. Kemp & Co., 14 South Square, Gray's  
Inn, London WC1R 5JJ, (GB)

**PATENT (CC, No, Kind, Date):** EP 1311949 A2 030521 (Basic)  
WO 2002029573 020411

**APPLICATION (CC, No, Date):** EP 2001965976 010817; WO 2001US25763 010817

**PRIORITY (CC, No, Date):** US 642061 000818

**DESIGNATED STATES:** AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI;  
LU; MC; NL; PT; SE; TR

**EXTENDED DESIGNATED STATES:** AL; LT; LV; MK; RO; SI

**INTERNATIONAL PATENT CLASS:** G06F-011/14

**NOTE:**

No A-document published by EPO

**LEGAL STATUS (Type, Pub Date, Kind, Text):**

Application: 021023 A2 International application. (Art. 158(1))

Application: 021023 A2 International application entering European  
phase

Application: 030521 A2 Published application without search report

Examination: 030521 A2 Date of request for examination: 20030317

**LANGUAGE (Publication,Procedural,Application):** English; English; English

5/5/2 (Item 2 from file: 348)  
DIALOG(R) File 348:EUROPEAN PATENTS  
(c) 2003 European Patent Office. All rts. reserv.

01425048

**MANIPULATION OF ZOMBIE FILES AND EVIL-TWIN FILES**

**MANIPULATION VON ZOMBIE- UND BOSE ZWILLINGS- DATEIEN**

**MANIPULATION DE FICHIERS ZOMBIES ET DE FICHIERS DIABOLIQUES**

**PATENT ASSIGNEE:**

Network Appliance, Inc., (2617421), 495 East Java Avenue, Sunnyvale,  
California 94089, (US), (Applicant designated States: all)

**INVENTOR:**

**CHEN, Ray** , 400 Castro Court, Campbell, CA 95008, (US)  
**EDWARDS, John, K.**, 1173 Crandano Court, Sunnyvale, CA 94087-2076, (US)  
**PATEL, Kayuri** , 20380 Stevens Creek Blvd., Apt. 219, Cupertino, CA 95014  
, (US)

**LEGAL REPRESENTATIVE:**

Leeming, John Gerard (74731), J.A. Kemp & Co., 14 South Square, Gray's  
Inn, London WC1R 5JJ, (GB)

**PATENT (CC, No, Kind, Date):** EP 1311948 A2 030521 (Basic)  
WO 2002019110 020307

**APPLICATION (CC, No, Date):** EP 2001964187 010817; WO 2001US25901 010817

**PRIORITY (CC, No, Date):** US 642066 000818

**DESIGNATED STATES:** AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI;  
LU; MC; NL; PT; SE; TR

**EXTENDED DESIGNATED STATES:** AL; LT; LV; MK; RO; SI

**INTERNATIONAL PATENT CLASS:** G06F-011/14

**NOTE:**

No A-document published by EPO

**LEGAL STATUS (Type, Pub Date, Kind, Text):**

Application: 020502 A2 International application. (Art. 158(1))  
Application: 020502 A2 International application entering European  
phase  
Application: 030521 A2 Published application without search report  
Examination: 030521 A2 Date of request for examination: 20030312  
LANGUAGE (Publication,Procedural,Application): English; English; English

5/5/3 (Item 3 from file: 348)  
DIALOG(R) File 348:EUROPEAN PATENTS  
(c) 2003 European Patent Office. All rts. reserv.

01422036

**IMPROVED SPACE ALLOCATION IN A WRITE ANYWHERE FILE SYSTEM**  
**SPEICHERBEREICHSUORDNUNG IN EINEM DATEISYSTEM ZUM BESCHREIBEN BELIEBIGER**  
**BEREICHE**

**ATTRIBUTION AMELIOREE D'ESPACE DANS UN SYSTEME WAFL (WRITE ANYWHERE FILE**  
**LAYOUT)**

PATENT ASSIGNEE:

Network Appliance, Inc., (2617422), 495 East Java Drive, Sunnyvale,  
California 94089, (US), (Applicant designated States: all)

INVENTOR:

DOUCETTE, Douglas, P., 1864 Scenic Avenue, Freeland, WA 98249, (US)

**LEWIS, Blake**, 2316 Greer Road, Palo Alto, CA 94303, (US)

EDWARDS, John, K., 1173 Crandano Court, Sunnyvale, CA 94087-2076, (US)

LEGAL REPRESENTATIVE:

Leeming, John Gerard (74731), J.A. Kemp & Co., 14 South Square, Gray's  
Inn, London WC1R 5JJ, (GB)

PATENT (CC, No, Kind, Date): EP 1311940 A2 030521 (Basic)  
WO 2002017057 020228

APPLICATION (CC, No, Date): EP 2001964151 010817; WO 2001US25822 010817

PRIORITY (CC, No, Date): US 642065 000818

DESIGNATED STATES: AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI;  
LU; MC; NL; PT; SE; TR

EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI

INTERNATIONAL PATENT CLASS: **G06F-003/06**

NOTE:

No A-document published by EPO

LEGAL STATUS (Type, Pub Date, Kind, Text):

Application: 020424 A2 International application. (Art. 158(1))

Application: 020424 A2 International application entering European  
phase

Application: 030521 A2 Published application without search report

Examination: 030521 A2 Date of request for examination: 20030312

LANGUAGE (Publication,Procedural,Application): English; English; English

5/5/4 (Item 1 from file: 349)  
DIALOG(R) File 349:PCT FULLTEXT  
(c) 2003 WIPO/Univentio. All rts. reserv.

00895454 \*\*Image available\*\*

**INSTANT SNAPSHOT**

**INSTANTANE**

Patent Applicant/Assignee:

NETWORK APPLIANCE INC, 495 East Java Drive, Sunnyvale, CA 94089, US, US  
(Residence), US (Nationality)

Inventor(s):

**LEWIS Blake**, 2316 Greer Road, Palo Alto, CA 94303, US,

EDWARDS John K, 1173 Crandano Court, Sunnyvale, CA 94087-2076, US,

VISWANATHAN Srinivasan, 751 Saltillo Place, Fremont, CA 94536, US

Legal Representative:

SWERNOFSKY Steven A (agent), Swernofsky Law Group, P.O. Box 390013,  
Mountain View, CA 94039-0013, US,

Patent and Priority Information (Country, Number, Date):

Patent: WO 200229573 A2-A3 20020411 (WO 0229573)

Application: WO 2001US25763 20010817 (PCT/WO US0125763)

Priority Application: US 2000642061 20000818  
(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE TR  
Main International Patent Class: **G06F-011/14**  
Publication Language: English  
Filing Language: English  
Fulltext Availability:  
Detailed Description  
Claims  
Fulltext Word Count: 6975

#### English Abstract

The invention provides an improved method and apparatus for creating a snapshot of a file system. In a first aspect of the invention, a "copy-on-write" mechanism is used. An effective snapshot mechanism must be efficient both in its use of storage space and in the time needed to create it because file systems are often large. The snapshot uses the same blocks as the active file system until the active file system is modified. Whenever a modification occurs, the modified data is copied to a new block and the old data is saved (henceforth called "copy-on-write"). In this way, the snapshot only uses space where it differs from the active file system, and the amount of work required to create the snapshot is small. In a second aspect of the invention, a record of which blocks are being used by the snapshot is included in the snapshot itself, allowing effectively instantaneous snapshot creation and deletion.

#### French Abstract

L'invention porte sur un procede et un appareil ameliores permettant de creer un instantane d'un systeme de fichier. Dans une premiere variante de l'invention, une logique de <= copie a l'ecriture >= est utilisee. Une logique d'instantane peut etre efficace lorsqu'elle est utilisee dans l'espace memoire et dans l'espace temps necessaire pour la cree, car les systemes de fichier sont souvent grands. L'instantane utilise les memes blocs que le systeme de fichier actif jusqu'a modification de ce dernier. Chaque fois qu'il y a une modification, les donnees modifiees sont copiees sur un nouveau bloc et les anciennes donnees sont sauvegardees (desormais appelee <= copie a l'ecriture >=). De cette facon, l'instantane n'utilise que l'espace ou il differe du systeme de fichier actif, et la quantite de travail requise pour creer l'instantane est faible. Dans une deuxieme variante de l'invention dont les blocs d'enregistrement sont utilises par l'instantane est inclus l'instantane lui-meme, ce qui permet de creer et d'annuler l'instantane de maniere efficace. Dans une troisieme variante, l'etat du systeme de fichier actif est defini par un ensemble de metafichiers ; notamment, un topogramme binaire (desormais appele <= topogramme actif >=) definit quels sont les blocs libres et quels sont ceux en utilisation. Le fichier inode definit quels sont les blocs utilises par chaque fichier, y compris les metafichiers. Le fichier inode lui-meme est defini par un <= root inode >= specifique, egalement connu sous le nom de <= bloc fsinfo >=. Le systeme commence a creer un nouvel instantane en faisant une copie de <= root inode >=. Cette copie de <= root inode >= devient la base de l'instantane. Root inode capture tous les etats requis pour creer l'instantane tels que l'emplacement de tous les fichiers et repertoires du systeme de fichier. Lors de mises a jour ulterieures du systeme de fichier actif, le systeme consulte les topogrammes binaires inclus dans l'instantane (<= le topogramme instantane >=) afin de determiner si un bloc libre peut etre reutilise ou s'il appartient a l'instantane. Cette logique permet au systeme de fichier actif de conserver une trace de ces blocs que chaque instantane utilise sans enregistrer d'informations supplementaires sur la tenue de la comptabilite du systeme de fichier. Dans une quatrieme variante, l'instantane peut etre egalement efface par simple suppression de son <= root inode >=. En outre, la tenue de la comptabilite n'est pas necessaire puisque l'instantane comprend sa propre definition. Dans une cinquieme variante, les charges de performance associees a la recherche de trois blocs sont reduites par l'inclusion d'un fichier recapitulatif qui identifie des blocs utilises par au moins un instantane ; c'est le OU logique de tous les fichiers de topogramme instantane. Le code d'affectation d'ecriture decide si un bloc est libre

en examinant le topogramme actif et le fichier recapitulatif. Le topogramme actif indique si le bloc est en cours d'utilisation dans le systeme de fichier actif. Le fichier recapitulatif indique si le bloc est utilise par un instantane quelconque. Dans une sixieme variante, le fichier recapitulatif indique si le bloc est mis a jour dans l'arriere-plan apres creation ou suppression d'un instantane. Ceci se produit simultanement avec d'autres operations du systeme de fichier. Deux bits sont mis en memoire dans le systeme de fichier "bloc fsinfo " pour chaque instantane. Ces deux bits indiquent si le fichier recapitulatif a besoin d'etre mis a jour a l'aide des informations du topogramme instantane suite a sa creation ou suppression. Lorsqu'un bloc est libere dans le systeme de fichier actif, le bloc correspondant du fichier recapitulatif est mis a jour avec le topogramme instantane a partir de l'instantane le plus recent, si cela n'a pas deja ete fait. Un topogramme binaire interne enregistre les mises a jour t

Legal Status (Type, Date, Text)

Publication 20020411 A2 Without international search report and to be republished upon receipt of that report.  
Examination 20020627 Request for preliminary examination prior to end of 19th month from priority date  
Search Rpt 20020711 Late publication of international search report  
Republication 20020711 A3 With international search report.  
Republication 20020711 A3 Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.  
Search Rpt 20020711 Late publication of international search report  
Claim Mod 20021114 Later publication of amended claims under Article 19 received: 20020702  
Republication 20021114 A3 With international search report.  
Republication 20021114 A3 With amended claims.

5/5/5 (Item 2 from file: 349)

DIALOG(R)File 349:PCT FULLTEXT

(c) 2003 WIPO/Univentio. All rts. reserv.

00885002 \*\*Image available\*\*

**MANIPULATION OF ZOMBIE FILES AND EVIL-TWIN FILES**

**MANIPULATION DE FICHIERS ZOMBIES ET DE FICHIERS DIABOLIQUES**

Patent Applicant/Assignee:

NETWORK APPLIANCE INC, 495 East Java Drive, Sunnyvale, CA 94089, US, US  
(Residence), US (Nationality)

Inventor(s):

**CHEN Ray** , 400 Castro Court, Campbell, CA 95008, US,  
**EDWARDS John K**, 1173 Crandano Court, Sunnyvale, CA 94087-2076, US,  
**PATEL Kayuri** , 20380 Stevens Creek Blvd., Apt. 219, Cupertino, CA 95014, US

Legal Representative:

SWERNOFSKY Steven A (agent), Swernofsky Law Group, P.O. Box 390013, Mountain View, CA 94039-0013, US,

Patent and Priority Information (Country, Number, Date):

Patent: WO 200219110 A2-A3 20020307 (WO 0219110)  
Application: WO 2001US25901 20010817 (PCT/WO US0125901)  
Priority Application: US 2000642066 20000818  
(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE TR

Main International Patent Class: **G06F-011/14**

Publication Language: English

Filing Language: English

Fulltext Availability:

Detailed Description

Claims

Fulltext Word Count: 6801

English Abstract

The invention provides a method and system for reliably performing extra-long operations in a reliable state-full system (such as a file



system). The system records consistency points, or otherwise assures reliability, notwithstanding the continuous performance of extra-long operations and the existence of intermediate states for those extra-long operations. Moreover, performance of extra-long operations is both deterministic and atomic with regard to consistency points (or other reliability techniques used by the system). The file system includes a separate portion of the file system reserved for files having extra-long operations in progress, including file deletion and file truncation. This separate portion of the file system is called the zombie filespace; it includes a separate name space from the regular ("live") file system that is accessible to users, and is maintained as part of the file system when recording a consistency point. The file system includes a file deletion manager that determines, before beginning any file deletion operation, whether it is necessary to first move the file being deleted to the zombie filespace. The file system includes a zombie file deletion manager that performs portions of the file deletion operation on zombie files in atomic units. The file system also includes a file truncation manager that determines, before beginning any file truncation operation, whether it is necessary to create a complementary file called an "evil twin". The truncation manager will move all blocks to be truncated from the file being truncated to the evil twin file. The file system includes a zombie file truncation manager that performs portions of the file truncation operation on the evil-twin file in atomic units. An additional advantage provided by the file system is that files having attached data elements, called "composite" files, can be subject to file deletion and other extra-long operations in a natural and reliable manner. The file system moves the entire composite file to the zombie filespace, deletes each attached data element individually, and thus resolves the composite file into a non-composite file. If the non-composite file is sufficiently small, the file deletion manager can delete the non-composite file without further need for the zombie filespace. However, if the non-composite file is sufficiently large, the file deletion manager can delete the non-composite file using the zombie filespace.

#### French Abstract

L'invention concerne un procede et un systeme permettant d'exécuter de maniere fiable des operations de tres longue duree dans un systeme fiable a integrite d'etat (tel qu'un systeme de fichiers). Le systeme enregistre des points de coherence, ou garantit d'une autre maniere la fiabilite, malgre l'exécution continue d'operations de tres longue duree et l'existence d'etats intermediaires destines a de telles operations. De plus, une execution d'operations de tres longue duree est aussi bien deterministe qu'atomique du point de vue des points de coherence (ou d'autres techniques de fiabilite utilisees par le systeme). Le systeme de fichiers comprend une partie distincte du systeme de fichiers reservee a des fichiers pour lesquels des operations de tres longue duree sont en cours, notamment la suppression et la troncature de fichiers. Cette partie distincte du systeme de fichiers est appelee l'espace-fichier zombie; celle-ci comprend un espace de nom distinct du systeme de fichiers normal (<=actif>=) qui est accessible par les utilisateurs, et est geree comme une partie du systeme de fichiers au moment de l'enregistrement d'un point de coherence. Le systeme de fichiers comprend un gestionnaire de suppression de fichiers determinant, avant le demarrage d'une suppression de fichier quelconque, s'il est necessaire de deplacer, dans un premier temps, le fichier en cours de suppression dans l'espace-fichier zombie. Le systeme de fichiers comprend un gestionnaire de suppression de fichiers zombies effectuant des phases de l'operation de suppression de fichiers sur les fichiers zombies dans des unites atomiques. Le systeme de fichiers comprend egalement un gestionnaire de troncature de fichiers determinant, avant le demarrage d'une operation de troncature quelconque, s'il est necessaire de creer un fichier complementaire appele <=jumeau diabolique>=. Le gestionnaire de troncature deplace tous les blocs devant etre tronques a partir du fichier en cours de troncature vers le fichier jumeau diabolique. Le systeme de fichiers comprend un gestionnaire de troncature de fichiers zombies effectuant des phases de l'operation de troncature de fichiers sur le fichier jumeau diabolique dans des unites atomiques. Le systeme de

fichiers presente un avantage supplementaire en ce que les fichiers comprenant des elements de donnees joints, appeles fichiers <=composites> pouvant etre soumis a une suppression de fichier ou d'autres operations de longue duree d'une maniere naturelle et fiable. Le systeme de fichiers deplace l'ensemble du fichier composite vers l'espace-fichier zombie, supprime, de maniere individuelle, chaque element de donnees joint et reduit, par consequent, le fichier composite en un fichier non composite. Si celui-ci est suffisamment petit, le gestionnaire de suppression de fichiers peut supprimer ce dernier sans devoir recourir a l'espace-fichier zombie. Cependant, si le fichier non composite est suffisamment grand, le gestionnaire de suppression de fichiers peut supprimer ledit fichier au moyen de l'espace-fichier zombie.

Legal Status (Type, Date, Text)

Publication 20020307 A2 Without international search report and to be republished upon receipt of that report.  
Search Rpt 20021031 Late publication of international search report  
Republication 20021031 A3 With international search report..  
Search Rpt 20021031 Late publication of international search report  
Examination 20030109 Request for preliminary examination prior to end of 19th month from priority date  
Claim Mod 20030220 Later publication of amended claims under Article 19 received: 20020923  
Republication 20030220 A3 With international search report.  
Republication 20030220 A3 With amended claims.

5/5/6 (Item 3 from file: 349)

DIALOG(R) File 349:PCT FULLTEXT

(c) 2003 WIPO/Univentio. All rts. reserv.

00882941 \*\*Image available\*\*

**IMPROVED SPACE ALLOCATION IN A WRITE ANYWHERE FILE SYSTEM**

**ATTRIBUTION AMELIOREE D'ESPACE DANS UN SYSTEME WAFL (WRITE ANYWHERE FILE LAYOUT)**

Patent Applicant/Assignee:

NETWORK APPLIANCE INC, 495 East Java Drive, Sunnyvale, CA 94089, US, US  
(Residence), US (Nationality)

Inventor(s):

DOUCETTE Douglas P, 1864 Scenic Avenue, Freeland, WA 98249, US,  
**LEWIS Blake**, 2316 Greer Road, Palo Alto, CA 94303, US,  
EDWARDS John K, 1173 Crandano Court, Sunnyvale, CA 94087-2076, US

Legal Representative:

SWERNOFSKY Steven A (agent), Swernofsky Law Group, P.O. Box 390013,  
Mountain View, CA 94039-0013, US,

Patent and Priority Information (Country, Number, Date):

Patent: WO 200217057 A2-A3 20020228 (WO 0217057)  
Application: WO 2001US25822 20010817 (PCT/WO US0125822)  
Priority Application: US 2000642065 20000818  
(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE TR

Main International Patent Class: **G06F-003/06**

International Patent Class: **G06F-011/14 ; G06F-012/02**

Publication Language: English

Filing Language: English

Fulltext Availability:

Detailed Description

Claims

Fulltext Word Count: 5623

English Abstract

The invention provides a method and system for improving data access of a reliable file system. In a first aspect of the invention, the file system determines the relative vacancy of a collection of storage blocks, herein called an "allocation area". This is accomplished by recording an array of binary numbers. Each binary number in the array describes the vacancy of a collection of storage blocks. The file system examines these binary

numbers when attempting to record file blocks in relatively contiguous areas on a storage medium, such as a hard disk. When a request to write to disk occurs, the system determines the average vacancy of all the allocation areas and queries the allocation areas for individual vacancy rates such as sequentially. The system preferably writes file blocks to the allocation areas that are above a threshold related to the average storage block vacancy of the file system. If the file in the request to write is larger than the selected allocation area, the next allocation area above the threshold is preferably used to write the remaining blocks of the file.

#### French Abstract

L'invention concerne un procede et un systeme permettant d'ameliorer l'accès a des donnees d'un systeme de fichiers fiable. Dans un premier mode de realisation de l'invention, le systeme de fichiers determine l'inoccupation relative d'un ensemble de blocs de stockage, appelee ici <=zone d'attribution>=, ceci est realise par l'enregistrement d'un reseau de nombres binaires. Chaque nombre binaire dans le reseau decrit l'inoccupation d'un ensemble de blocs de stockage. Le systeme de fichiers examine ces nombres binaires au cours de la tentative d'enregistrement de blocs de fichiers dans des zones relativement contigues sur un support de stockage, tel qu'un disque dur. Lorsqu'une demande d'écriture sur disque apparait, le systeme determine l'inoccupation moyenne de toutes les zones d'attribution et execute une requete concernant le taux d'inoccupation individuel des zones d'attribution de maniere sequentielle. Le systeme ecrit, de preference, des blocs de fichiers dans des zones d'attribution qui sont au-dessus d'un seuil relatif a l'inoccupation moyenne des blocs de stockage du systeme de fichiers. Si dans la demande d'écriture, le fichier est plus important que la zone d'attribution selectionnee, la zone d'attribution suivante au-dessus du seuil est, de preference, utilisee en vue d'ecrire les blocs restants du fichier.

#### Legal Status (Type, Date, Text)

Publication 20020228 A2 Without international search report and to be  
republished upon receipt of that report.  
Examination 20021017 Request for preliminary examination prior to end of  
19th month from priority date  
Search Rpt 20030320 Late publication of international search report  
Republication 20030320 A3 With international search report.

5/5/7 (Item 1 from file: 350)

DIALOG(R) File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

014956760 \*\*Image available\*\*

WPI Acc No: 2003-017274/200301

XRPX Acc No: N03-013168

**Power control method in network data processing system, involves sending logical partition power off request to logical partitions service processor, if supervisor decides that additional active partition is present**

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC ); IBM UK LTD (IBMC )

Inventor: DAWKINS G J; LEE V H; **PATEL K** ; PHAN P D; WILLOUGHBY D R;

WILLOUGHBY D

Number of Countries: 100 Number of Patents: 002

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 20020124194	A1	20020905	US 2001798167	A	20010301	200301 B
WO 200271215	A2	20020912	WO 2002GB867	A	20020227	200301

Priority Applications (No Type Date): US 2001798167 A 20010301

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
-----------	------	-----	----	----------	--------------

US 20020124194	A1		16	G06F-001/26	
----------------	----	--	----	-------------	--

WO 200271215	A2 E			G06F-009/445	
--------------	------	--	--	--------------	--

Designated States (National): AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA

CH CN CO CR CU CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HR HU ID IL IN  
IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ  
OM PH PL PT RO RU SD SE SG SI SK SL TJ TM TN TR TT TZ UA UG UZ VN YU ZA  
ZM ZW

Designated States (Regional): AT BE CH CY DE DK EA ES FI FR GB GH GM GR  
IE IT KE LS LU MC MW MZ NL OA PT SD SE SL SZ TR TZ UG ZM ZW

Abstract (Basic): US 20020124194 A1

NOVELTY - On receiving logical partition power off request from the partition operating system, the request is passed to a supervisor. If the supervisor decides that additional active partition is present in the data processing system, logical partition power off request is sent to the service processor, handling processors assigned to the partitions, else, system power off request is sent to the service processor for complete shutdown.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for the following:

- (1) Data processing system; and
- (2) Computer program product for power control in data processing system.

USE - For controlling power in network data processing system having several logical partitions.

ADVANTAGE - Provides an ability to reboot or reset processors assigned to a particular logical partition without rebooting or resetting other processors assigned to other logical partitions in the data processing system. Allows for handling resources for a logical partition individually without affecting other logical partitions.

DESCRIPTION OF DRAWING(S) - The figure shows a profile of distributed data processing system.

pp; 16 DwgNo 1/11

Title Terms: POWER; CONTROL; METHOD; NETWORK; DATA; PROCESS; SYSTEM; SEND; LOGIC; PARTITION; POWER; REQUEST; LOGIC; PARTITION; SERVICE; PROCESSOR; SUPERVISION; DECIDE; ADD; ACTIVE; PARTITION; PRESENT

Derwent Class: T01

International Patent Class (Main): G06F-001/26 ; G06F-009/445

International Patent Class (Additional): G06F-001/28 ; G06F-001/30

File Segment: EPI

5/5/8 (Item 2 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

014929517 \*\*Image available\*\*

WPI Acc No: 2002-750226/200281

XRPX Acc No: N02-590890

**virtual NVRAM access method for data processing system, involves providing access to requested portion of NVRAM, if virtual address in request is within range of addresses for which processor is authorized to access**

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC )

Inventor: LEE V H; PATEL K ; WILLOUGHBY D R

Number of Countries: 001 Number of Patents: 002

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 20020129212	A1	20020912	US 2001798292	A	20010301	200281 B
US 6567897	B2	20030520	US 2001798292	A	20010301	200336

Priority Applications (No Type Date): US 2001798292 A 20010301

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
US 20020129212	A1	9	G06F-012/14	
US 6567897	B2		G06F-012/00	

Abstract (Basic): US 20020129212 A1

NOVELTY - A request including a virtual address is received by the NVRAM from a processor. If the virtual address is within the range of addresses for which the processor is authorized to access, then the

access to the requested portion of the NVRAM is provided to the processor.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for the following:

(1) Computer program product for virtual NVRAM access facility; and

(2) virtual NVRAM access system.

USE - For accessing virtual NVRAM for logically partitioned (LPAR) data processing system.

ADVANTAGE - Enforces logical partitioning of NVRAM to which multiple partitions with the data processing system have access.

DESCRIPTION OF DRAWING(S) - The figure shows the flowchart illustrating the virtual NVRAM accessing process.

pp; 9 DwgNo 3/3

Title Terms: VIRTUAL; ACCESS; METHOD; DATA; PROCESS; SYSTEM; ACCESS;

REQUEST; PORTION; VIRTUAL; ADDRESS; REQUEST; RANGE; ADDRESS; PROCESSOR;

AUTHORISE; ACCESS

Derwent Class: T01; U14

International Patent Class (Main): G06F-012/00 ; G06F-012/14

File Segment: EPI

5/5/9 (Item 3 from file: 350)

DIALOG(R) File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

014769679 \*\*Image available\*\*

WPI Acc No: 2002-590383/200263

XRPX Acc No: N02-468597

**Firmware updating method for complex computer, involves determining whether system component has current level of firmware and accordingly updating firmware in background operation**

Patent Assignee: IBM CORP (IBM )

Inventor: KEHNE K G; MEHTA C; PATEL J M; PATEL K

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 20020092008	A1	20020711	US 2000726290	A	20001130	200263 B

Priority Applications (No Type Date): US 2000726290 A 20001130

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20020092008	A1		9	G06F-009/455	

US 20020092008 A1 9 G06F-009/455

Abstract (Basic): US 20020092008 A1

NOVELTY - A level of firmware on a system component is compared with a current level of the firmware stored on a memory. The firmware stored in the system component is updated in a background operation, when the system component does not have current level of the firmware, while the system remains available to the user for other actions.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for the following:

(1) Computer program product storing firmware updating instructions; and

(2) Firmware updating system.

USE - For updating firmwares of complex computer used as web server in multiprocessing environment.

ADVANTAGE - Updates system firmware in a background operation, thereby allowing the user to utilize the computer for other purposes during update process.

DESCRIPTION OF DRAWING(S) - The figure shows a flowchart illustrating firmware updating process.

pp; 9 DwgNo 3/3

Title Terms: FIRMWARE; UPDATE; METHOD; COMPLEX; COMPUTER; DETERMINE; SYSTEM

; COMPONENT; CURRENT; LEVEL; FIRMWARE; ACCORD; UPDATE; FIRMWARE;

BACKGROUND; OPERATE

Derwent Class: T01; U14

International Patent Class (Main): G06F-009/455  
File Segment: EPI

5/5/10 (Item 4 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2003 Thomson Derwent. All rts. reserv.

014762461 \*\*Image available\*\*  
WPI Acc No: 2002-583165/200262  
Related WPI Acc No: 2002-339696  
XRPX Acc No: N02-462528

**File system operating method for file server system, involves recording changes to the zombie file space of a file system in a persistent memory**

Patent Assignee: CHEN R C (CHEN-I); EDWARDS J (EDWA-I); PATEL K (PATE-I)

Inventor: CHEN R C; EDWARDS J; PATEL K

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 20020083081	A1	20020627	US 2000642066	A	20000818	200262 B
			US 2001932579	A	20010817	

Priority Applications (No Type Date): US 2001932579 A 20010817; US  
2000642066 A 20000818

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
US 20020083081	A1	14	G06F-012/00	CIP of application US 2000642066

Abstract (Basic): US 20020083081 A1

NOVELTY - The method involves recording changes to the zombie file space of a file system in a persistent memory.

USE - For file server system.

ADVANTAGE - Enables reliable execution of extra-long operations in a file system.

DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of a portion of a file system.

pp; 14 DwgNo 1/5

Title Terms: FILE; SYSTEM; OPERATE; METHOD; FILE; SERVE; SYSTEM; RECORD;  
CHANGE; FILE; SPACE; FILE; SYSTEM; PERSISTENT; MEMORY

Derwent Class: T01

International Patent Class (Main): G06F-012/00

File Segment: EPI

5/5/11 (Item 5 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2003 Thomson Derwent. All rts. reserv.

014762452 \*\*Image available\*\*  
WPI Acc No: 2002-583156/200262  
Related WPI Acc No: 2002-435358  
XRPX Acc No: N02-462519

**File and directory content capturing method for file system, involves refraining from writing data in storage blocks in response to active map or copy of active map recorded in file system**

Patent Assignee: NETWORK APPLIANCE INC (NETW-N)

Inventor: EDWARDS J; LEWIS B ; VISWANATHAN S

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 20020083037	A1	20020627	US 2000642061	A	20000818	200262 B
			US 2001932578	A	20010817	

Priority Applications (No Type Date): US 2001932578 A 20010817; US  
2000642061 A 20000818

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
-----------	------	--------	----------	--------------

US 20020083037 A1 14 G06F-007/00 CIP of application US 2000642061  
Abstract (Basic): US 20020083037 A1

NOVELTY - Active map is recorded in the file system of storage blocks not available for writing data. Consistency point including a copy of the active map is recorded in the file system. Writing of data in the storage blocks is refrained in response to the active map and the copy of the active map.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for the following:

- (1) Previous version saving method;
- (2) File system operating method;
- (3) Summary map updating method; and
- (4) Active map updating method.

USE - File and directory content capturing method for file system of data storage system.

ADVANTAGE - Enables creating and deleting snapshots quickly and capture the contents of the files and directories in the file system, quickly and efficiently.

DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of the instant snapshot system.

pp; 14 DwgNo 1/5

Title Terms: FILE; DIRECTORY; CONTENT; CAPTURE; METHOD; FILE; SYSTEM;  
WRITING; DATA; STORAGE; BLOCK; RESPOND; ACTIVE; MAP; COPY; ACTIVE; MAP;  
RECORD; FILE; SYSTEM

Derwent Class: T01

International Patent Class (Main): G06F-007/00

File Segment: EPI

5/5/12 (Item 6 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

014614654 \*\*Image available\*\*

WPI Acc No: 2002-435358/200246

Related WPI Acc No: 2002-583156

XRPX Acc No: N02-342702

**Capturing contents of the files and directories in a file system comprising a set of storage blocks in mass storage by recording consistency point in file system including a consistent version of file system at a previous time**

Patent Assignee: NETWORK APPLIANCE INC (NETW-N)

Inventor: EDWARDS J K; LEWIS B ; VISWANATHAN S

Number of Countries: 020 Number of Patents: 002

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 200229573	A2	20020411	WO 2001US25763	A	20010817	200246 B
EP 1311949	A2	20030521	EP 2001965976	A	20010817	200334
			WO 2001US25763	A	20010817	

Priority Applications (No Type Date): US 2000642061 A 20000818

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
-----------	------	-----	----	----------	--------------

WO 200229573	A2	E	27	G06F-011/14	
--------------	----	---	----	-------------	--

Designated States (Regional): AT BE CH CY DE DK ES FI FR GB GR IE IT LU  
MC NL PT SE TR

EP 1311949	A2	E		G06F-011/14	Based on patent WO 200229573
------------	----	---	--	-------------	------------------------------

Designated States (Regional): AT BE CH CY DE DK ES FI FR GB GR IE IT LI  
LU MC NL PT SE TR

Abstract (Basic): WO 200229573 A2

NOVELTY - An active map in a file system of storage blocks not available for writing data is recorded with a consistency point in the file system including a consistent version of the file system at a previous time. The consistency point includes a copy of the active map at the previous time. Data from writing to storage blocks is refrained in response to the active map and at least one of the copy of the

active map at the previous time.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for:

- (a) a method of saving previous versions of an active file system including the contents of directories in the file system
- (b) a method of operating a file system
- (c) in a file system including an active map of information indicating in-use and free blocks for maintaining a set of snapshots

USE - For creating a snapshot of a file system.

ADVANTAGE - Provides an improved technique for more quickly and efficiently capturing the contents of the files and directories in the file system at a particular point in time.

DESCRIPTION OF DRAWING(S) - The drawing shows a block diagram of an instant snapshot.

pp; 27 DwgNo 2/5

Title Terms: CAPTURE; CONTENT; FILE; DIRECTORY; FILE; SYSTEM; COMPRISE; SET  
; STORAGE; BLOCK; MASS; STORAGE; RECORD; CONSISTENCY; POINT; FILE; SYSTEM  
; CONSISTENT; VERSION; FILE; SYSTEM; TIME

Derwent Class: T01

International Patent Class (Main): G06F-011/14

File Segment: EPI

5/5/13 (Item 7 from file: 350)

DIALOG(R) File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

014541379

WPI Acc No: 2002-362082/200239

SRPX Acc No: N02-283040

**Computer implemented method of allocating space in a write anywhere data store using a map of unavailable store blocks to determine a value for regions of store indicating available storage space**

Patent Assignee: NETWORK APPLIANCE INC (NETW-N)

Inventor: DOUCETTE D P; EDWARDS J K; LEWIS B

Number of Countries: 020 Number of Patents: 002

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 200217057	A2	20020228	WO 2001US25822	A	20010817	200239 B
EP 1311940	A2	20030521	EP 2001964151	A	20010817	200334
			WO 2001US25822	A	20010817	

Priority Applications (No Type Date): US 2000642065 A 20000818

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes

WO 200217057 A2 E 25 G06F-003/06

Designated States (Regional): AT BE CH CY DE DK ES FI FR GB GR IE IT LU  
MC NL PT SE TR

EP 1311940 A2 E G06F-003/06 Based on patent WO 200217057

Designated States (Regional): AT BE CH CY DE DK ES FI FR GB GR IE IT LI  
LU MC NL PT SE TR

Abstract (Basic): WO 200217057 A2

NOVELTY - The value represents the relative vacancy of a collection of storage blocks by recording an array of binary numbers each describing the vacancy of the collection of storage blocks. When attempting to record files in relatively contiguous areas of the store the file system examines the values and determines the average vacancy of all the collections of storage blocks and writes in areas where the values are above a threshold related to the average vacancy of the whole system. If the file being written is larger than the selected collection of store blocks, the next collection of blocks with a value above the threshold is used to store the remainder of the file, and so on as necessary.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for

- (a) a method of improved space allocation in a write anywhere file system
- (b) and apparatus for improved space allocation in a write anywhere



file system

USE - In data storage systems.

ADVANTAGE - Improved technique for finding relatively large free areas of data storage efficiently.

pp; 25 DwgNo 0/2

Title Terms: COMPUTER; IMPLEMENT; METHOD; ALLOCATE; SPACE; WRITING; DATA; STORAGE; MAP; UNAVAILABLE; STORAGE; BLOCK; DETERMINE; VALUE; REGION; STORAGE; INDICATE; AVAILABLE; STORAGE; SPACE

Derwent Class: T01; T03

International Patent Class (Main): G06F-003/06

File Segment: EPI

5/5/14 (Item 8 from file: 350)

DIALOG(R) File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

014539594 \*\*Image available\*\*

WPI Acc No: 2002-360297/200239

XRFX Acc No: N02-281423

**Detecting method for environmental faults in redundant components of a computer system using registered failure condition as data in an architected error log**

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC )

Inventor: KITAMORN A; MCLAUGHLIN C A; PATEL K ; THORSON D L

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 6345369	B1	20020205	US 98190559	A	19981112	200239 B

Priority Applications (No Type Date): US 98190559 A 19981112

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6345369	B1		13	G06F-011/16	

Abstract (Basic): US 6345369 B1

NOVELTY - The method involves monitoring system environment conditions, including a status for redundant power supply and cooling components. A failure condition is registered with an appropriate error type when a monitored system environment condition exceeds a design threshold.

DETAILED DESCRIPTION - The registered failure condition is used as data in an architected error log. A standard EPOW (environmental and power warning) arrangement is extended to handle the complexity of the redundant power and cooling components being monitored.

INDEPENDENT CLAIMS are included for

(1) an apparatus for detecting environmental faults of redundant components in a computer system

(2) a method for supporting detection of environmental faults of redundant components in a computer system.

USE - For detecting environmental and power problems in computer system.

ADVANTAGE - Improved environmental and power warning system.

DESCRIPTION OF DRAWING(S) - The figure shows the operation of an EPOW interrupt.

pp; 13 DwgNo 7/8

Title Terms: DETECT; METHOD; ENVIRONMENT; FAULT; REDUNDANT; COMPONENT; COMPUTER; SYSTEM; REGISTER; FAIL; CONDITION; DATA; ERROR; LOG

Derwent Class: T01; U21

International Patent Class (Main): G06F-011/16

File Segment: EPI

5/5/15 (Item 9 from file: 350)

DIALOG(R) File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

014522310     \*\*Image available\*\*  
WPI Acc No: 2002-343013/200238  
XRPX Acc No: N02-269765

**Data back-up method for computer network, involves transmitting portion of each flagged file to remote location for backup, as attachment to message**

Patent Assignee: BLACK J G (BLAC-I); HUGHES R C (HUGH-I); LEWIS B (LEWI-I); PURNELL M W (PURN-I)

Inventor: BLACK J G; HUGHES R C; **LEWIS B** ; PURNELL M W

Number of Countries: 001    Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
GB 2365154	A	20020213	GB 200017865	A	20000720	200238    B

Priority Applications (No Type Date): GB 200017865 A 20000720

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
GB 2365154	A	26	G06F-011/14	

Abstract (Basic): GB 2365154 A

NOVELTY - Scanning and flagging of files of a host computer are continuously performed. A portion of each flagged file is transmitted to a remote location for backup, as an attachment to a message which has header. The header specifies sufficient information for systematic handling of the flagged file at the remote location.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(a) Computer program product including recorded medium storing back-up program;

(b) Backup service computer for storing backup data

USE - For computer network e.g. private network such as local area network (LAN) and public network such as Internet.

ADVANTAGE - Backup services are commercially performed at low cost and with no complication. It runs as part of the shutdown sequence, so extra time is not required at the end of the day to run a backup process. Hence lengthy backup process such as overnight backing up is completely avoided. Multiple copying of messages adds almost no complexity to the backup process, highly reliable in house backup service at minimal cost and management overhead.

DESCRIPTION OF DRAWING(S) - The figure shows backup system of data in computer network.

pp; 26 DwgNo 1/5

Title Terms: DATA; BACK; UP; METHOD; COMPUTER; NETWORK; TRANSMIT; PORTION; FLAG; FILE; REMOTE; LOCATE; ATTACH; MESSAGE

Derwent Class: T01; U21

International Patent Class (Main): **G06F-011/14**

File Segment: EPI

5/5/16        (Item 10 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2003 Thomson Derwent. All rts. reserv.

014518993     \*\*Image available\*\*  
WPI Acc No: 2002-339696/200237  
Related WPI Acc No: 2002-583165  
XRPX Acc No: N02-267123

**File system operating method where file system includes live file space accessible to users and zombie file space not accessible to users recording changes to zombie file space in persistent memory**

Patent Assignee: NETWORK APPLIANCE INC (NETW-N)

Inventor: CHEN R; EDWARDS J K; **PATEL K**

Number of Countries: 020    Number of Patents: 002

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 200219110	A2	20020307	WO 2001US25901	A	20010817	200237    B
EP 1311948	A2	20030521	EP 2001964187	A	20010817	200334

Priority Applications (No Type Date): US 2000642066 A 20000818

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes

WO 200219110 A2 E 27 G06F-011/00

Designated States (Regional): AT BE CH CY DE DK ES FI FR GB GR IE IT LU  
MC NL PT SE TR

EP 1311948 A2 E G06F-011/14 Based on patent WO 200219110

Designated States (Regional): AT BE CH CY DE DK ES FI FR GB GR IE IT LI  
LU MC NL PT SE TR

Abstract (Basic): WO 200219110 A2

NOVELTY - The method involves recording changes to a zombie file space in a persistent memory. The method further involves transferring a file from live file space to the zombie file space. Links associating disk blocks with the file are broken in several steps while the file is associated with the zombie file space. The recording of changes includes recording the breaking of links in several steps. The live file space is altered to reflect the deletion operation.

USE - For file server systems in which it is desired to maintain file system consistency.

ADVANTAGE - Provides technique for extra-long operations in a reliable state-full system (such as a file system) that is not subject to known drawbacks.

DESCRIPTION OF DRAWING(S) - The figure shows a file structure in a system using a zombie file space.

pp; 27 DwgNo 2/5

Title Terms: FILE; SYSTEM; OPERATE; METHOD; FILE; SYSTEM; LIVE; FILE; SPACE  
; ACCESS; USER; FILE; SPACE; ACCESS; USER; RECORD; CHANGE; FILE; SPACE;  
PERSISTENT; MEMORY

Derwent Class: T01

International Patent Class (Main): G06F-011/00 ; G06F-011/14

File Segment: EPI

5/5/17 (Item 11 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

014301484 \*\*Image available\*\*

WPI Acc No: 2002-122188/200216

XRPX Acc No: N02-091663

**Integrated electronic shopping cart system for e-commerce applications,  
has affiliated websites associated servers which provide catalogs of item  
information to main server for processing item order**

Patent Assignee: IPRINT.COM INC (IPRI-N); BELL R (BELL-I); HODSON D  
(HODS-I); PATEL K (PATE-I); RUBIN M (RUBI-I)

Inventor: BELL R; HODSON D; PATEL K ; RUBIN M

Number of Countries: 091 Number of Patents: 003

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 200195208	A1	20011213	WO 2001US17674	A	20010601	200216 B
AU 200175082	A	20011217	AU 200175082	A	20010601	200225
US 20020052806	A1	20020502	US 2000208860	P	20000602	200234
			US 2001872514	A	20010601	

Priority Applications (No Type Date): US 2000208860 P 20000602; US  
2001872514 A 20010601

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes

WO 200195208 A1 E 79 G06F-017/60

Designated States (National): AE AL AM AT AU AZ BA BB BG BR BY CA CH CN  
CR CU CZ DE DK DM EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP  
KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX NO NZ PL PT RO RU SD SE  
SG SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW

Designated States (Regional): AT BE CH CY DE DK EA ES FI FR GB GH GM GR

IE IT KE LS LU MC MW MZ NL OA PT SD SE SL SZ TR TZ UG ZW  
AU 200175082 A G06F-017/60 Based on patent WO 200195208  
US 20020052806 A1 G06F-017/60 Provisional application US 2000208860

Abstract (Basic): WO 200195208 A1

NOVELTY - Affiliated servers (31n) associated with affiliated websites, provide catalogs of item information to a main server (31) associated with a website. The main server includes shopping cart and catalog applications to maintain order of items selected from affiliated sites. The main server processes the shopping cart order and provides order fulfillment information to affiliated servers for subsequent processing.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(a) Method of incorporating integrated shopping cart functionality in computer network;

(b) Integrated electronic shopping cart program

USE - Electronic shopping cart system integrated with Internet or world wide web for business-to-business and business-to-consumer applications in e-commerce.

ADVANTAGE - Since the proprietor of one website offers various products/services from a multitude of different affiliated websites, increased number of users access the website. Hence that website is benefited by this integrated shopping cart functionality. Also, since the e-commerce functionality of the integrated shopping cart is performed by the main server, the affiliated websites need not include their own e-commerce functionality. Hence maintenance costs and overhead costs with respect to including an e-commerce software engine are reduced. Hence the affiliated websites are benefited by this integrated shopping cart functionality.

DESCRIPTION OF DRAWING(S) - The figure shows the schematic diagram of an integrated shopping cart system.

Main server (31)

Affiliated servers (31n)

pp; 79 DwgNo 2/13

Title Terms: INTEGRATE; ELECTRONIC; SHOPPING; CART; SYSTEM; APPLY;  
ASSOCIATE; SERVE; CATALOGUE; ITEM; INFORMATION; MAIN; SERVE; PROCESS;  
ITEM; ORDER

Derwent Class: T01

International Patent Class (Main): G06F-017/60

File Segment: EPI

5/5/18 (Item 12 from file: 350)

DIALOG(R) File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

014140835 \*\*Image available\*\*

WPI Acc No: 2001-625046/200172

XRPX Acc No: N01-465806

Multiple-tier intelligent bus arbitration method involves replacing low tier initiating device to high tier, when a pending transaction is recognized in low tier initiating device

Patent Assignee: DING C (DING-I); PATEL K (PATE-I); REKEITA D W (REKE-I); TEXAS INSTR INC (TEXI )

Inventor: DING C; PATEL K ; REKEITA D W

Number of Countries: 001 Number of Patents: 002

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 20010027505	A1	20011004	US 9760467	P	19970930	200172 B
			US 98162975	A	19980929	
			US 2001796016	A	20010228	
US 6393508	B1	20020521	US 9760467	P	19970930	200239
			US 98162975	A	19980929	
			US 2001796016	A	20010228	

Priority Applications (No Type Date): US 9760467 P 19970930; US 98162975 A

19980929; US 2001796016 A 20010228

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20010027505	A1	12	G06F-012/00	Provisional application US 9760467	

US 6393508	B1	G06F-012/00	Cont of application US 98162975
			Provisional application US 9760467
			Cont of application US 98162975

Abstract (Basic): US 20010027505 A1

NOVELTY - An initiating device (30) in lower tier accesses the secondary peripheral component interconnect (PCI) bus (42) and attempts transaction with primary PCI bus (26) through a PCI to PCI bridge (38). When pending in transaction is recognized, the initiating device is replaced to higher tier and transaction is completed. The device is then changed to lower tier, after transaction completion.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

- (a) PCI to PCI bridge arbiter;
- (b) PCI to PCI bridge

USE - For intelligent bus arbitration in peripheral component interconnect (PCI) to PCI bridge.

ADVANTAGE - Since secondary PCI bus internal arbiter can determine a low tier device having a pending transaction and place the device into high tier, transaction can be completed more quickly.

DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of typical PCI to PCI bus applications.

PCI buses (26,42)  
Initiating device (30)  
PCI to PCI bridge (38)  
pp; 12 DwgNo 1/7

Title Terms: MULTIPLE; TIER; INTELLIGENCE; BUS; ARBITER; METHOD; REPLACE;  
LOW; TIER; INITIATE; DEVICE; HIGH; TIER; PENDING; TRANSACTION; RECOGNISE;  
LOW; TIER; INITIATE; DEVICE

Derwent Class: T01

International Patent Class (Main): G06F-012/00

International Patent Class (Additional): G06F-013/14 ; G06F-013/38

File Segment: EPI

5/5/19 (Item 13 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

014086365 \*\*Image available\*\*

WPI Acc No: 2001-570579/200164

XRPX Acc No: N01-425228

**Memory allocation method for data processing system, involves receiving memory request which is responded, based on which memory is allocated**

Patent Assignee: SUN MICROSYSTEMS INC (SUNM )

Inventor: BOUCHER M; HINKER P; LEWIS B

Number of Countries: 019 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 200161498	A1	20010823	WO 2001US4682	A	20010214	200164 B

Priority Applications (No Type Date): US 2000504876 A 20000216

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 200161498	A1	E	19	G06F-012/02	

Designated States (Regional): AT BE CH CY DE DK ES FI FR GB GR IE IT LU  
MC NL PT SE TR

Abstract (Basic): WO 200161498 A1

NOVELTY - A memory request is received and is transmitted to block of memory. A request is responded and memory is allocated based on memory request.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

- (a) Memory accessing method for data processing system;
- (b) System for allocating memory;
- (c) Data processing system for accessing memory;
- (d) Computer readable medium which includes instructions for allocating memory

USE - For data processing systems to access and allocate memory.

ADVANTAGE - Multiprocessor computing provides maximum efficiency and operational benefits in single or multithreaded, single processor environment. Distributive and adaptive features of smart-alloc allows minimization of number of accesses of operating system and yields increased system performance.

DESCRIPTION OF DRAWING(S) - The figure shows the details of general access tree and fast access tree of memory access and allocation system.

pp; 19 DwgNo 2/4

Title Terms: MEMORY; ALLOCATE; METHOD; DATA; PROCESS; SYSTEM; RECEIVE;

MEMORY; REQUEST; RESPOND; BASED; MEMORY; ALLOCATE

Derwent Class: T01

International Patent Class (Main): G06F-012/02

File Segment: EPI

5/5/20 (Item 14 from file: 350)

DIALOG(R) File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

014073447

WPI Acc No: 2001-557660/200162

XRPX Acc No: N01-414415

**Method of allocating memory to a program in a data processor by responding to instructions within the program to send pointers allocating memory space to respective program threads**

Patent Assignee: SUN MICROSYSTEMS INC (SUNM )

Inventor: BOUCHER M; HINKER P; LEWIS B

Number of Countries: 093 Number of Patents: 002

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 200161471	A2	20010823	WO 2001US4681	A	20010214	200162 B
AU 200136989	A	20010827	AU 200136989	A	20010214	200176

Priority Applications (No Type Date): US 2000504877 A 20000216

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes

WO 200161471 A2 E 20 G06F-009/00

Designated States (National): AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CR CU CZ DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW

Designated States (Regional): AT BE CH CY DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW MZ NL OA PT SD SE SL SZ TR TZ UG ZW

AU 200136989 A G06F-009/00 Based on patent WO 200161471

Abstract (Basic): WO 200161471 A2

NOVELTY - Each thread in a program itself requests memory access and receives a pointer allocating memory space to that thread. The process is undertaken essentially independently of the operating system and allows threads to access memory virtually simultaneously. A linked list of references to allocated memory is maintained and memory space is allocated and de-allocated as required.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for

(a) computer readable medium containing instructions for allocating memory space to program threads

(b) a system for allocating memory space to program threads

(c) and a system for de-allocating memory space to program threads

USE - In data processing systems.

ADVANTAGE - Allows program threads to access a memory virtually simultaneously by means of a function included in the program, independent of the operating system which provides a memory call function.

pp; 20 DwgNo 0/5

Title Terms: METHOD; ALLOCATE; MEMORY; PROGRAM; DATA; PROCESSOR; RESPOND; INSTRUCTION; PROGRAM; SEND; POINT; ALLOCATE; MEMORY; SPACE; RESPECTIVE; PROGRAM; THREAD

Derwent Class: T01

International Patent Class (Main): G06F-009/00

File Segment: EPI

5/5/21 (Item 15 from file: 350)

DIALOG(R) File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

013895889 \*\*Image available\*\*

WPI Acc No: 2001-380102/200140

XRPX Acc No: N01-278595

Intelligent configuration register access on PCI-PCI bridge involves determining whether reading of configuration data should be trapped or not and returning configuration data when reading should be trapped

Patent Assignee: TEXAS INSTR INC (TEXI )

Inventor: DICKENS D E; PATEL K ; REKEITA D W

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 6189063	B1	20010213	US 9760508	A	19970930	200140 B
			US 98162702	A	19980929	

Priority Applications (No Type Date): US 9760508 P 19970930; US 98162702 A 19980929

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6189063	B1	10	G06F-009/445	Provisional application	US 9760508

Abstract (Basic): US 6189063 B1

NOVELTY - The method involves identifying a reading of configuration data of a connected peripheral component interconnect (PCI) device (30) and determining whether the read should be trapped or not. The type of PCI device is determined for the connected PCI device and configuration data are returned in a desired format, when a read should be trapped, otherwise read is passed to other connected PCI devices.

DETAILED DESCRIPTION - An INDEPENDENT CLAIM is also included for a PCI to PCI bridge apparatus.

USE - For intelligent configuration register access on PCI-PCI bridges.

ADVANTAGE - Provides cost effective and intelligent scheme for configuration register access that eliminates overhead e.g. cost, board area and complex design for implementing and loading of configuration registers. Allows PCI-PCI bridge to handle configuration register access for existing PCI devices, thus allowing existing PCI devices to run under new system.

DESCRIPTION OF DRAWING(S) - The figure is a block diagram of a data storage associated with PCI-PCI bridge and a connected PCI device.

PCI device (30)

pp; 10 DwgNo 5/6

Title Terms: INTELLIGENCE; CONFIGURATION; REGISTER; ACCESS; BRIDGE; DETERMINE; READ; CONFIGURATION; DATA; TRAP; RETURN; CONFIGURATION; DATA; READ; TRAP

Derwent Class: T01

International Patent Class (Main): G06F-009/445

International Patent Class (Additional): G06F-012/00 ; G06F-013/10

File Segment: EPI

5/5/22 (Item 16 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2003 Thomson Derwent. All rts. reserv.

013494414 \*\*Image available\*\*  
WPI Acc No: 2000-666355/200065  
XRPX Acc No: N00-493886

**Control flow program execution method for data driven multiprocessor computer systems by defining memory region, dividing into multiple blocks and using dependencies to indicate relationship between blocks**

Patent Assignee: SUN MICROSYSTEMS INC (SUNM )  
Inventor: BRADLEY L; LEWIS B  
Number of Countries: 027 Number of Patents: 003  
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
EP 1026585	A2	20000809	EP 2000400305	A	20000203	200065 B
JP 2000285084	A	20001013	JP 200026991	A	20000204	200101
US 6378066	B1	20020423	US 99244138	A	19990204	200232

Priority Applications (No Type Date): US 99244138 A 19990204

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 1026585	A2	E	35	G06F-009/44	
Designated States (Regional): AL AT BE CH CY DE DK ES FI FR GB GR IE IT					
LI LT LU LV MC MK NL PT RO SE SI					
JP 2000285084	A		84	G06F-015/16	
US 6378066	B1			G06F-009/305	

Abstract (Basic): EP 1026585 A2

NOVELTY - The output of blocks (1,1), (2,1), (3,1), (4,1) sharing the first state are required before the sequential processing of each block sharing the second state and blocks sharing the third state are processed.

DETAILED DESCRIPTION - An INDEPENDENT CLAIM is also included for a computer implemented method for developing a data flow program, a data processing system containing a development tool (user interface), a computer program stored on a computer readable medium.

USE - For data driven multiprocessor computer systems.

ADVANTAGE - It can be implemented as a combination of hardware and/or software in both object and non-object oriented programming systems.

DESCRIPTION OF DRAWING(S) - The figure shows a directed acyclic graph illustrating the dependency relationship among blocks of the memory region.

First Block State ((1,1), (2,1), (3,1), (4,1))  
pp; 35 DwgNo 4/19

Title Terms: CONTROL; FLOW; PROGRAM; EXECUTE; METHOD; DATA; DRIVE;  
MULTIPROCESSOR; COMPUTER; SYSTEM; DEFINE; MEMORY; REGION; DIVIDE;  
MULTIPLE; BLOCK; INDICATE; RELATED; BLOCK

Derwent Class: T01

International Patent Class (Main): G06F-009/305 ; G06F-009/44 ;  
G06F-015/16

International Patent Class (Additional): G06F-009/06 ; G06F-009/445 ;  
G06F-009/45 ; G06F-009/455 ; G06F-009/46

File Segment: EPI

5/5/23 (Item 17 from file: 350)  
DIALOG(R)File 350:Derwent WPIX  
(c) 2003 Thomson Derwent. All rts. reserv.

013494412 \*\*Image available\*\*  
WPI Acc No: 2000-666353/200065  
XRPX Acc No: N00-493884

**Computer implemented method for developing data flow program in multiprocessor computer systems (data driven) by defining memory region**



and dividing into multiple blocks reflecting designated portion of  
program based on dependencies

Patent Assignee: SUN MICROSYSTEMS INC (SUNM )

Inventor: LEWIS B

Number of Countries: 027 Number of Patents: 003

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
EP 1026583	A2	20000809	EP 2000400303	A	20000203	200065 B
JP 2000284950	A	20001013	JP 200024087	A	20000201	200101
US 6389587	B1	20020514	US 99244136	A	19990204	200239

Priority Applications (No Type Date): US 99244136 A 19990204

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
-----------	------	-----	----	----------	--------------

EP 1026583	A2	E	31	G06F-009/44	
------------	----	---	----	-------------	--

Designated States (Regional): AL AT BE CH CY DE DK ES FI FR GB GR IE IT

LI LT LU LV MC MK NL PT RO SE SI

JP 2000284950	A		84	G06F-009/06	
---------------	---	--	----	-------------	--

US 6389587	B1			G06F-009/44	
------------	----	--	--	-------------	--

Abstract (Basic): EP 1026583 A2

NOVELTY - The output of all the blocks (1,1), (2,1), (3,1), (4,1) sharing the first state are required for processing by each of the blocks sharing the second state and the blocks sharing the second state must be processed before each of the three groups of the three blocks that share the third state are processed.

DETAILED DESCRIPTION - An INDEPENDENT CLAIM is also included for a data processing system containing graphical user interface, a computer program stored on a computer readable medium

USE - For developing data flow program in multiprocessor computer systems.

ADVANTAGE - It can be implemented as a combination of hardware and/or software in both object and non-object oriented programming systems.

DESCRIPTION OF DRAWING(S) - The figure shows a directed acyclic graph illustrating the dependency relationship among blocks of the memory region.

First State Blocks ((1,1), (2,1), (3,1), (4,1))

pp; 31 DwgNo 4/17

Title Terms: COMPUTER; IMPLEMENT; METHOD; DEVELOP; DATA; FLOW; PROGRAM; MULTIPROCESSOR; COMPUTER; SYSTEM; DATA; DRIVE; DEFINE; MEMORY; REGION; DIVIDE; MULTIPLE; BLOCK; REFLECT; DESIGNATED; PORTION; PROGRAM; BASED

Derwent Class: T01

International Patent Class (Main): G06F-009/06 ; G06F-009/44

International Patent Class (Additional): G06F-009/46 ; G06F-015/16 ;

G06F-015/82

File Segment: EPI

5/5/24 (Item 18 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

013476708 \*\*Image available\*\*

WPI Acc No: 2000-648651/200063

Related WPI Acc No: 2002-489538

XRPX Acc No: N00-480878

Thread execution analyzing performance method for multiprocessor computer systems by receiving and displaying data reflecting state of thread execution during measurement period

Patent Assignee: SUN MICROSYSTEMS INC (SUNM ); BOUCHER M (BOUC-I); DENNIE S (DENN-I); LEWIS B (LEWI-I); WEEK J (WEEK-I)

Inventor: BOUCHER M; DENNIE S; LEWIS B ; WEEK J

Number of Countries: 027 Number of Patents: 004

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
EP 1026592	A2	20000809	EP 2000400307	A	20000203	200063 B

JP 2000235511 A 20000829 JP 200024065 A 20000201 200063  
US 20020073360 A1 20020613 US 99244895 A 19990204 200255  
US 6434714 B1 20020813 US 99244895 A 19990204 200255

Priority Applications (No Type Date): US 99244895 A 19990204

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes

EP 1026592 A2 E 10 G06F-011/36

Designated States (Regional): AL AT BE CH CY DE DK ES FI FR GB GR IE IT

LI LT LU LV MC MK NL PT RO SE SI

JP 2000235511 A 29 G06F-011/34

US 20020073360 A1 G06F-011/30

US 6434714 B1 G06F-011/00

Abstract (Basic): EP 1026592 A2

NOVELTY - Processor (220) executes multiple threads (212-218) in parallel and an operating system partitions memory (240) into segments designated for operations of each thread. Each segment comprises of state identifiers, developer specified information and thread identification information during execution of program a thread leaves a state. Developer specified data represents the performance analysis data collected.

DETAILED DESCRIPTION - An INDEPENDENT CLAIM is also included for a system for collecting and displaying performance data associated with threads executing in a multi-processor computing system.

USE - For multiprocessor computer systems

ADVANTAGE - The developer can determine the context in which a performance problem occurs and provides an insight into diagnosing the problem.

DESCRIPTION OF DRAWING(S) - The figure shows a block diagram of performance analysis system.

Multiple Thread (212-218)

Processor (220)

Memory (240)

pp; 10 DwgNo 2/4

Title Terms: THREAD; EXECUTE; PERFORMANCE; METHOD; MULTIPROCESSOR; COMPUTER ; SYSTEM; RECEIVE; DISPLAY; DATA; REFLECT; STATE; THREAD; EXECUTE; MEASURE; PERIOD

Derwent Class: T01

International Patent Class (Main): G06F-011/00 ; G06F-011/30 ; G06F-011/34 ; G06F-011/36

International Patent Class (Additional): G06F-003/00

File Segment: EPI

5/5/25 (Item 19 from file: 350)

DIALOG(R) File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

012192951 \*\*Image available\*\*

WPI Acc No: 1998-609864/199851

XRPX Acc No: N98-474453

Automatic enhanced report generator for computer system - operates in response to data not used in original report to allow changes in appearance of original report

Patent Assignee: XEROX CORP (XERO )

Inventor: HAXTON K C; PATEL K ; TRIPATHI A

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 5832504	A	19981103	US 94237578	A	19940503	199851 B
			US 96684988	A	19960722	

Priority Applications (No Type Date): US 94237578 A 19940503; US 96684988 A 19960722

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes

Abstract (Basic): US 5832504 A

The enhanced report generator is loaded into the memory of a computer that contains an original report generator responsive to the data for generating an original report. The enhanced report generator is responsive to the data that is not used in the original report.

A set of rules (15) is stored for each type of enhanced report produced by the generator. Consequently, appearance of the original report is changed by modifying colour, size, arrangement or spacing of alphanumeric content or graphics to generate enhanced report.

ADVANTAGE - Adapts to computer that use different languages and hardware. Enables use of less number of object oriented enhancement programs to cover wide range of printing applications. Allows additional level of data security for e.g. pay roll data. Enables different categories of users to access different fractions of total database.

Dwg.1/3

Title Terms: AUTOMATIC; ENHANCE; REPORT; GENERATOR; COMPUTER; SYSTEM;  
OPERATE; RESPOND; DATA; ORIGINAL; REPORT; ALLOW; CHANGE; APPEAR; ORIGINAL  
; REPORT

Derwent Class: T01

International Patent Class (Main): G06F-017/30

File Segment: EPI

Set	Items	Description
S1	667734	RESERV??? OR SAVE OR SAVING OR RETAIN??? OR (SET OR PUT) () - (ASIDE OR APART) OR PREARRANGE?
S2	10176	(UNALLOCATED OR "NOT"(1W) (ALLOCATE? OR ASSIGN? OR EARMARK? OR ALLOT??)OR UNUSED OR FREE) () (BLOCK? OR CHUNK? OR BITS OR B- BYTES OR SEGMENT? OR PIECE? OR PART? OR SPACE)
S3	2966	FILE()SYSTEM?
S4	2564534	EQUAL OR CORRESPOND? OR MATCH? OR PARALLEL? OR PARITY OR A- LIGNMENT
S5	2815	(NUMBER OR AMOUNT OR TOTAL OR SUM) () (BLOCK? OR CHUNK? OR B- ITS OR BYTES OR SEGMENT? OR PIECE? OR PART? OR SPACE)
S6	3373845	NEED? OR SIZE? OR CAPACIT? OR EXTENT OR LIMIT? OR BOUNDAR? OR THRESHOLD OR REQUIRE?
S7	3	WRITE()ANYWHERE()FILE()SYSTEM OR WAFL
S8	51	S1 (3N) S2
S9	0	S8 AND S3
S10	2	S1 AND S2 AND S3
S11	32	S1 AND S5 AND S6
S12	0	S2 AND S4 AND S5 AND S6
S13	0	S2 AND S11
S14	7	S4 AND S11
S15	12	S7 OR S10 OR S14

File 347:JAPIO Oct 1976-2003/Feb(Updated 030603)

(c) 2003 JPO & JAPIO

File 350:Derwent WPIX 1963-2003/UD,UM &UP=200337

(c) 2003 Thomson Derwent

15/5/1 (Item 1 from file: 347)  
DIALOG(R)File 347:JAPIO  
(c) 2003 JPO & JAPIO. All rts. reserv.

04921837 \*\*Image available\*\*  
PARTS FEEDING METHOD AND ITS DEVICE

PUB. NO.: 07-214437 [JP 7214437 A]  
PUBLISHED: August 15, 1995 (19950815)  
INVENTOR(s): SAWARA YOSHIO  
KITAGAWA HAJIME  
APPLICANT(s): DAIKIN IND LTD [000285] (A Japanese Company or Corporation),  
JP (Japan)  
APPL. NO.: 06-011833 [JP 9411833]  
FILED: February 03, 1994 (19940203)  
INTL CLASS: [6] B23P-021/00; B23P-019/00; B65G-047/14  
JAPIO CLASS: 25.2 (MACHINE TOOLS -- Cutting & Grinding); 26.9  
(TRANSPORTATION -- Other); 36.1 (LABOR **SAVING** DEVICES --  
Industrial Robots  
JAPIO KEYWORD: R098 (ELECTRONIC MATERIALS -- Charge Transfer Elements, CCD &  
BBD)

#### ABSTRACT

PURPOSE: To improve the average parts taking out number by setting the upper **limit** number of parts to be taken out continuously from a parts feeder whose parts taking out time is long in comparison with a parts feeder whose parts taking out time is the shortest.

CONSTITUTION: At a second feeder 2 (a side whose parts taking out time is longer than that of a first feeder 1), parts that can be taken out are searched by the image processing portion of a feeder controller 31, and the number of parts that can be taken out is discriminated. Next, whether or not the number thus discriminated is smaller than the upper **limit** number of parts set beforehand is discriminated. In a case in which a smaller number is discriminated, parts are taken out in regard to all of the discriminated number, and a series of processing is finished. Meanwhile, in a case in which it is discriminated that this discriminated number is not smaller than the upper **limit** number set beforehand ( **equal** to the upper **limit** number or more than the upper **limit** number ), parts enough only for the upper **limit** number are taken out, and a series of processing is finished.

15/5/2 (Item 2 from file: 347)  
DIALOG(R)File 347:JAPIO  
(c) 2003 JPO & JAPIO. All rts. reserv.

02957858 \*\*Image available\*\*  
WINDING OF CYLINDRICAL ARMATURE WINDING

PUB. NO.: 01-255458 [JP 1255458 A]  
PUBLISHED: October 12, 1989 (19891012)  
INVENTOR(s): MIYAMOTO TADAHIRO  
SATO YUKINORI  
APPLICANT(s): YASKAWA ELECTRIC MFG CO LTD [000662] (A Japanese Company or Corporation), JP (Japan)  
APPL. NO.: 63-081851 [JP 8881851]  
FILED: April 02, 1988 (19880402)  
INTL CLASS: [4] H02K-015/04  
JAPIO CLASS: 43.1 (ELECTRIC POWER -- Generation); 36.1 (LABOR **SAVING** DEVICES -- Industrial Robots  
JOURNAL: Section: E, Section No. 870, Vol. 14, No. 9, Pg. 166, January 10, 1989 (19890110)

#### ABSTRACT

PURPOSE: To improve the **total space** factor of winding, by dividing a multilayer element coil into an arbitrary number of belt coils and changing

the **size** of wires in accordance with the difference of respective peripheral lengths.

CONSTITUTION: Outside element coils 1-3 and inside element coils 1-3 for respective phases are prepared at first. Wires having slightly larger **sizes** are employed for the outside element coils 1-3 for respective phases and the length (l) of the element coil is designed so as to be slightly longer. Next, outside and inside belt coils 4 are finished so as to be cylindrical coils. The inside cylindrical coil 21 is inserted into the finished outside cylindrical coil 20. In this case, the inside cylindrical coil is inserted into the outside cylindrical coil 20 while being deviated so that the coil ends of respective cylindrical coils will never be superposed, the polarities of respective magnetomotive forces of the outside and inside cylindrical coils are in the same direction and the positions of the maximum magnetomotive force points become same. Finally, the terminals of respective phases are connected in series or in **parallel** to form a balanced armature coil and, thereafter, the coil is impregnated with resin whereby the production of the cylindrical coil may be finished.

15/5/3 (Item 3 from file: 347)  
DIALOG(R) File 347:JAPIO  
(c) 2003 JPO & JAPIO. All rts. reserv.

01829255 \*\*Image available\*\*  
PARTS DATA CONTROL SYSTEM USING VARIABLE LENGTH INDEX

PUB. NO.: 61-043355 [JP 61043355 A]  
PUBLISHED: March 01, 1986 (19860301)  
INVENTOR(s): MORIMOTO MASAYUKI  
APPLICANT(s): FUJITSU LTD [000522] (A Japanese Company or Corporation), JP  
(Japan)  
APPL. NO.: 59-164634 [JP 84164634]  
FILED: August 06, 1984 (19840806)  
INTL CLASS: [4] G06F-012/00; G06F-003/06  
JAPIO CLASS: 45.2 (INFORMATION PROCESSING -- Memory Units); 45.3  
(INFORMATION PROCESSING -- Input Output Units)  
JOURNAL: Section: P, Section No. 477, Vol. 10, No. 202, Pg. 43, July  
15, 1986 (19860715)

#### ABSTRACT

PURPOSE: To secure a flexible countermeasure to the increase of devices and also to attain high-speed access and the **save** of file **capacity**, by controlling plural pieces of device information with just a single random access file.

CONSTITUTION: A data control part 2 of a parts data control system 1 starts a data register part 3, a data deletion part 4, a data replacement part 5, a data retrieval part 6, etc. according to a request given from a terminal containing a display 7 and a keyboard 8. Thus the part 2 performs control in response to the request given from the terminal. A random access file 10 is **equal** to a data memory file as well as a data set used for a magnetic disk device, for example, that can write data directly on or read the data out of a designated block. A device index part 11, and a produce **number** **block** 12 and a chart **number** **block** 13 are provided to the file 10.

15/5/7 (Item 4 from file: 350)  
DIALOG(R) File 350:Derwent WPIX  
(c) 2003 Thomson Derwent. All rts. reserv.

014614654 \*\*Image available\*\*  
WPI Acc No: 2002-435358/200246  
Related WPI Acc No: 2002-583156  
XRPX Acc No: N02-342702

**Capturing contents of the files and directories in a file system  
comprising a set of storage blocks in mass storage by recording**

consistency point in file system including a consistent version of  
file system at a previous time

Patent Assignee: NETWORK APPLIANCE INC (NETW-N)  
Inventor: EDWARDS J K; LEWIS B; VISWANATHAN S  
Number of Countries: 020 Number of Patents: 002  
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 200229573	A2	20020411	WO 2001US25763	A	20010817	200246 B
EP 1311949	A2	20030521	EP 2001965976	A	20010817	200334
			WO 2001US25763	A	20010817	

Priority Applications (No Type Date): US 2000642061 A 20000818

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 200229573	A2	E	27	G06F-011/14	
Designated States (Regional): AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE TR					
EP 1311949	A2	E		G06F-011/14	Based on patent WO 200229573
Designated States (Regional): AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR					

Abstract (Basic): WO 200229573 A2

NOVELTY - An active map in a file system of storage blocks not available for writing data is recorded with a consistency point in the file system including a consistent version of the file system at a previous time. The consistency point includes a copy of the active map at the previous time. Data from writing to storage blocks is refrained in response to the active map and at least one of the copy of the active map at the previous time.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for:

(a) a method of saving previous versions of an active file system including the contents of directories in the file system  
(b) a method of operating a file system  
(c) in a file system including an active map of information indicating in-use and free blocks for maintaining a set of snapshots

USE - For creating a snapshot of a file system .

ADVANTAGE - Provides an improved technique for more quickly and efficiently capturing the contents of the files and directories in the file system at a particular point in time.

DESCRIPTION OF DRAWING(S) - The drawing shows a block diagram of an instant snapshot.

pp; 27 DwgNo 2/5

Title Terms: CAPTURE; CONTENT; FILE; DIRECTORY; FILE; SYSTEM; COMPRISE; SET  
; STORAGE; BLOCK; MASS; STORAGE; RECORD; CONSISTENCY; POINT; FILE; SYSTEM  
; CONSISTENT; VERSION; FILE; SYSTEM; TIME

Derwent Class: T01

International Patent Class (Main): G06F-011/14

File Segment: EPI

15/5/8 (Item 5 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

014541379

WPI Acc No: 2002-362082/200239

XRPX Acc No: N02-283040

Computer implemented method of allocating space in a write anywhere data store using a map of unavailable store blocks to determine a value for regions of store indicating available storage space

Patent Assignee: NETWORK APPLIANCE INC (NETW-N)  
Inventor: DOUCETTE D P; EDWARDS J K; LEWIS B  
Number of Countries: 020 Number of Patents: 002  
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 200217057	A2	20020228	WO 2001US25822	A	20010817	200239 B

EP 1311940 A2 20030521 EP 2001964151 A 20010817 200334  
WO 2001US25822 A 20010817

Priority Applications (No Type Date): US 2000642065 A 20000818

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes

WO 200217057 A2 E 25 G06F-003/06

Designated States (Regional): AT BE CH CY DE DK ES FI FR GB GR IE IT LU  
MC NL PT SE TR

EP 1311940 A2 E G06F-003/06 Based on patent WO 200217057

Designated States (Regional): AT BE CH CY DE DK ES FI FR GB GR IE IT LI  
LU MC NL PT SE TR

Abstract (Basic): WO 200217057 A2

NOVELTY - The value represents the relative vacancy of a collection of storage blocks by recording an array of binary numbers each describing the vacancy of the collection of storage blocks. When attempting to record files in relatively contiguous areas of the store the file system examines the values and determines the average vacancy of all the collections of storage blocks and writes in areas where the values are above a threshold related to the average vacancy of the whole system. If the file being written is larger than the selected collection of store blocks, the next collection of blocks with a value above the threshold is used to store the remainder of the file, and so on as necessary.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for

(a) a method of improved space allocation in a **write anywhere file system**

(b) and apparatus for improved space allocation in a **write anywhere file system**

USE - In data storage systems.

ADVANTAGE - Improved technique for finding relatively large free areas of data storage efficiently.

pp; 25 DwgNo 0/2

Title Terms: COMPUTER; IMPLEMENT; METHOD; ALLOCATE; SPACE; WRITING; DATA; STORAGE; MAP; UNAVAILABLE; STORAGE; BLOCK; DETERMINE; VALUE; REGION; STORAGE; INDICATE; AVAILABLE; STORAGE; SPACE

Derwent Class: T01; T03

International Patent Class (Main): G06F-003/06

File Segment: EPI

15/5/10 (Item 7 from file: 350)

DIALOG(R)File 350:Derwent WPIX

(c) 2003 Thomson Derwent. All rts. reserv.

011843975 \*\*Image available\*\*

WPI Acc No: 1998-260885/199823

Related WPI Acc No: 1999-633517

XRPX Acc No: N98-205712

**Source file conversion method to target file system - involves saving data and meta data of source file system in target files of target file system and parsing data associated with files of source file system out of target files**

Patent Assignee: MICROSOFT CORP (MICR-N)

Inventor: MILEWSKI B B; SHOROFF S

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 5742818	A	19980421	US 95573479	A	19951215	199823 B

Priority Applications (No Type Date): US 95573479 A 19951215

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes

US 5742818 A 14 G06F-017/30

Abstract (Basic): US 5742818 A



The method of converting a source **file system**, including meta data and data files, to a target **file system** involves creating target **file system** meta information. The data files and meta data of the source **file system** are saved in one or more target files in the target **file system**.

Data associated with files of the source **file system** are parsed out of the one or more target files. The parsed data is provided to the target **file system** with instructions to the target **file system** to **save** the parsed data as files in the target **file system**.

ADVANTAGE - Maintains data integrity. Simplifies programming necessary for converting disk from source **file system** to target **file system** without requiring backup of source file on auxiliary peripheral storage device, or requiring large amount of **free space** to be available on disk.

Dwg.7/9b

Title Terms: SOURCE; FILE; CONVERT; METHOD; TARGET; FILE; SYSTEM; **SAVE** ;  
DATA; META; DATA; SOURCE; FILE; SYSTEM; TARGET; FILE; TARGET; FILE;  
SYSTEM; PARSE; DATA; ASSOCIATE; FILE; SOURCE; FILE; SYSTEM; TARGET; FILE  
Derwent Class: T01  
International Patent Class (Main): G06F-017/30  
File Segment: EPI

15/5/11 (Item 8 from file: 350)  
DIALOG(R) File 350:Derwent WPIX  
(c) 2003 Thomson Derwent. All rts. reserv.

011480022 \*\*Image available\*\*  
WPI Acc No: 1997-457927/199743  
XRPX Acc No: N97-381419

Write anywhere **file - system layout e.g. for file server back-up -**  
**always writes new data to unallocated blocks on disk, creates snapshots**  
**for duplicate inode to be virtual read-only copies of file system**

Patent Assignee: NETWORK APPLIANCE CORP (NETW-N)  
Inventor: HITZ D; LAU J; MALCOLM M; RAKITZIS B  
Number of Countries: 001 Number of Patents: 001  
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
CA 2165912	A	19970622	CA 2165912	A	19951221	199743 B

Priority Applications (No Type Date): CA 2165912 A 19951221

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
CA 2165912	A		93	G06F-017/30	

Abstract (Basic): CA 2165912 A

The file system progresses from one consistent state to another, with changes to the file system tightly controlled. The set of self-consistent blocks on disk that is rooted by the root inode is referred to as a consistency point. To implement consistency points, new data is written to unallocated blocks on disk. A new consistency point occurs when the fsinfo block (2440) is updated by writing a new root inode for the inode file (1210) into it. Thus, as long as the root inode is not updated, the state of the file system represented on disk does not change.

Snapshots are created that are read-only copies of the file system. A snapshot uses no disk space when it is initially created. It is designed so that many different snapshots can be created for the same file system. A multi-bit free-block map file (1630) is used to prevent data from being overwritten on disk.

USE/ADVANTAGE - Keeps file system in consistent state and creates read-only copies of file system. Unlike previous file systems that create clone by duplicating entire inode file and all indirect blocks, only inode that describes inode file is duplicated. Actual disk space required for snapshot is only 128 bytes for duplicated inode.

Dwg.2/23b

Title Terms: WRITING; FILE; SYSTEM; LAYOUT; FILE; SERVE; BACK; UP; WRITING;

NEW; DATA; BLOCK; DISC; SNAPSHOT; DUPLICATE; VIRTUAL; READ; COPY; FILE;  
SYSTEM

Derwent Class: T01

International Patent Class (Main): G06F-017/30

File Segment: EPI

Set	Items	Description
S1	435664	RESERV??? OR SAVE OR SAVING OR RETAIN??? OR (SET OR PUT)()- (ASIDE OR APART) OR PREARRANGE?
S2	14226	(UNALLOCATED OR "NOT"(1W) (ALLOCATE? OR ASSIGN? OR EARMARK? OR ALLOT??)OR UNUSED OR FREE) () (BLOCK? OR CHUNK? OR BITS OR B- BYTES OR SEGMENT? OR PIECE? OR PART? OR SPACE)
S3	4856	FILE()SYSTEM?
S4	1167631	EQUAL OR CORRESPOND? OR MATCH? OR PARALLEL? OR PARITY OR A- LIGNMENT
S5	7389	(NUMBER OR AMOUNT OR TOTAL OR SUM) () (BLOCK? OR CHUNK? OR B- ITS OR BYTES OR SEGMENT? OR PIECE? OR PART? OR SPACE)
S6	1321044	NEED? OR SIZE? OR CAPACIT? OR EXTENT OR LIMIT? OR BOUNDAR? OR THRESHOLD OR REQUIRE?
S7	258	WRITE()ANYWHERE()FILE()SYSTEM OR WAFL
S8	106	S1 (3N) S2
S9	3	S8 (S) S3
S10	29	S1 (S) S2 (S) S3
S11	31073	S1 (S) S4 (S) S6
S12	42	S2 (S) S4 (S) S5 (S) S6
S13	67	S9 OR S10 OR S12
S14	41	S13 AND IC=G06F?
S15	17	S14 AND IC=(G06F-012? OR G06F-017?)
S16	24	S14 NOT S15

File 348:EUROPEAN PATENTS 1978-2003/Jun W01

(c) 2003 European Patent Office

File 349:PCT FULLTEXT 1979-2002/UB=20030605,UT=20030529

(c) 2003 WIPO/Univentio

UNPUBLISHED -- RIGHTS **RESERVED** UNDER THE COPYRIGHT  
LAWS OF THE UNITED STATES. USE OF A COPYRIGHT NOTICE  
IS PRECAUTIONARY ONLY AND...User: root  
Host: bunny  
Class: bunny  
Job: stdin  
0001GI  
Copyright (C) 1995 MITSUBISHI ELECTRIC ITA. ALL RIGHTS **RESERVED** .

UNPUBLISHED -- RIGHTS **RESERVED** UNDER THE COPYRIGHT  
LAWS OF THE UNITED STATES. USE OF A COPYRIGHT NOTICE  
IS PRECAUTIONARY ONLY AND...to time-ou  
return(o);  
000168  
I  
000 169  
Copyright(C)1995 MITSUBISHI ELECTRIC ITA. ALL RIGHTS **RESERVED** .

UNPUBLISHED -- RIGHTS **RESERVED** UNDER THE COPYRIGHT  
LAWS OF THE UNITED STATES. USE OF A COPYRIGHT NOTICE  
IS PRECAUTIONARY ONLY AND...kp",flags);  
Say(11RAMQCS: ALL DONE");  
return(O);  
000173  
Copyright(C)199S MITSUBISHI ELECTRIC ITA. ALL RIGHTS **RESERVED** .

UNPUBLISHED -- RIGHTS **RESERVED** UNDER THE COPYRIGHT  
LAWS OF THE UNITED STATES. USE OF A COPYRIGHT NOTICE  
IS PRECAUTIONARY ONLY AND...

16/5,K/18 (Item 6 from file: 349)  
DIALOG(R)File 349:PCT FULLTEXT  
(c) 2003 WIPO/Univentio. All rts. reserv.

00300850 \*\*Image available\*\*  
**UPDATE MECHANISM FOR COMPUTER STORAGE CONTAINER MANAGER**  
**MOYEN DE MISE A JOUR POUR MODULE DE GESTION D'ELEMENTS DE STOCKAGE**  
**D'ORDINATEURS**  
Patent Applicant/Assignee:  
APPLE COMPUTER INC,  
Inventor(s):  
HARRIS Jared M,  
RUBEN Ira L,  
Patent and Priority Information (Country, Number, Date):  
Patent: WO 9519001 A1 19950713  
Application: WO 95US196 19950104 (PCT/WO US9500196)  
Priority Application: US 94177853 19940105  
Designated States: AM AT AU BB BG BR BY CA CH CN CZ DE DK EE ES FI GB GE HU  
JP KE KG KP KR KZ LK LR LT LU LV MD MG MN MW MX NL NO NZ PL PT RO RU SD  
SE SI SK TJ TT UA UZ VN KE MW SD SZ AT BE CH DE DK ES FR GB GR IE IT LU  
MC NL PT SE BF BJ CF CG CI CM GA GN ML MR NE SN TD TG  
Main International Patent Class: **G06F-009/44**  
Publication Language: English  
Fulltext Availability:  
Detailed Description  
Claims  
Fulltext Word Count: 119635

#### English Abstract

Methods and data structures are defined which permit information to be stored as objects in target containers and update containers. A target container defines a first state of the information, and the update container, which can point to the target container, identifies changes to the information in the first state which would be sufficient to update the first information state to a second information state. Update

containers may be nested to any depth. When an application program opens an update container, the procedure searches down the chain until it finds the ultimate target container. It then creates in-memory structures for providing access to the objects and value data represented in such container. The procedure then works its way back up the chain, performing the changes on the in-memory structure, which are called for in each of the update containers.

#### French Abstract

L'invention se rapporte a des structures de donnees et a des procedes permettant de stocker des informations sous forme d'objets dans des elements de stockage cibles et dans des elements de stockage de mise a jour. Un element de stockage cible definit un premier etat relatif aux informations, et l'element de stockage de mise a jour, qui peut identifier l'element de stockage cible, identifie des modifications des informations presentant le premier etat, qui devraient permettre la mise a jour dudit premier etat en un second etat. Les elements de stockage de mise a jour peuvent s'emboiter indefiniment. Lorsqu'un programme d'application ouvre un element de stockage de mise a jour, la procedure appliquee consiste a effectuer une recherche le long de la chaine jusqu'a ce que l'element de stockage cible au bout de la chaine soit identifie. Des structures en memoire sont alors creees afin de permettre l'accès aux objets et aux donnees de valeur representes dans un tel element de stockage. La procedure consiste alors a remonter la chaine, et a effectuer, dans la structure en memoire, les modifications requises dans chacun des elements de stockage de mise a jour.

Main International Patent Class: **G06F-009/44**

Fulltext Availability:

Claims

#### Claim

... object. The ID is also an immediate value. Note that the generation numbers of the TOC entries **correspond** to generation numbers of the relevant content objects, The generation of the TOC value itself is the...be accessed linearly in time and keeps the objects in the required sorted order. The index tables **correspond** to "powers" of a chosen index table size. For example, if the table size is 256 and...StdObjID  
TOC  
Del CM StdObjID TOC Type total value data space  
eted deleted  
CM  
.@tdObjID) TOC Fre **free space** list as value  
70 - CM StdObjID TOC@ Type  
e segments  
go CM  
StdObjID.TOC  
Ne CM StdObjID...

...The starting object ID seed for the container as a constant. For a new container it is **equal** to CMTOCSeedGlobalName. When updating, this is the initial seed for the updating container as derived from the...words, offset 0 to the offset of the last byte of this container's label).  
CMTOCDeletedGlobalName/CMTOCValueTypeGlobalName - **total space** deleted  
A CM ULONG value representing all deleted value data space created by CMDeleteValueData0,  
CMTOCTotalFreeGlobalName/CMMCValueTypeGlobalName - **free space** list  
An internal list of immediate value data segments, each of whose offset and size represent available **free space**, Note, the total space represent here may be less than CMTOCDeletedGlobalName since not all space

is recorded due...

...container when updating. For an appended target, the type is always CMTargetContainerName and the value's offset/ **size** are to the appended target

(including the label). This is similar to embedded containers. For separate updaters...is not a dynamic value, then the routine merely obtains the real value from the real value **segments** (e.g. 426) and assembles them into the caller's buffer (step 806). It then re

turns...case (3) the old segment must be split and the new segment inserted between the two old **segments**. The original logical offsets of the old segments are preserved. This becomes important only for deletions, but...not by accident). Thus the code buffers the updating information and calls CMReadValueData or CMWriteValueData only when **needed**, The buffer **size** is defined in a header file as UpdateBufSize. Note that this same buffered I/O is supported...

...the I/O handlers. A header file controls this by defining the input and output TOC buffer **sizes** (TOCInputBufSize and TOCOutputBufSize, respectively). Defining either or both of these as 0 turns off the corresponding TOC...However, there's a "chicken-and-egg" problem here with respect to object IDs. The dynamic value **needs** to be created along with all its associated objects to get at the target, but ...infos, data edits, moves, etc.). The value data for such a property are all the updating instructions **needed** to bring all the values the associated object "up to date". These instructions cannot be applied until...

...the updating CCB, After reading in the non-private TOC, the touched chain will represent all objects **needing** updating. The touched chain can now be walked much like close-time processing to process the updating...state ready to record new updates. Note, that while loading the TOC entries into memory, the logical **size** and offsets are generated for the value headers and segments. Applying updates at this time changes the logical **sizes** and offsets. Thus, after each value's updates are completed, if the updates involved data editing, the...

...can be seen that these instructions have parameters that can be directly used in many of the **corresponding** API calls, If update recording wasn't suppressed, a loop condition would result, with objects being added...

...release of the data.

H. Open-time Processing for Multi-layered Updaters  
The previous discussion was mainly **limited** to one container updating another, It is fairly simple extension to the algorithm to allow for multiple...provided for the purposes of illustration and description, It is not intended to be exhaustive or to **limit** the invention to the precise forms disclosed. Obviously, many

modifications and variations will be apparent to practitioners...

...an implementation of the routines

may construct only those aspects of the TOC in-memory which are **needed** for a particular operation after an update container is opened. The embodiments described herein were chosen and...endif

CM-CFUNCTIONS

/\* The following generates a segment directive for Mac only due to 32K Jump Table \*/ **Limitations** . If you don't know what I'm talking about don't worry about it. The\*/ /\* default...

...the info container there. The prime piece of info, among other things is the TOC offset and **size** so we can load in the TOC. 3 5 This function returns true and returns the Label...CONTAINERNAME, Q, i3, i4, 5);

return (faLsj);

2 0

\*majorVersion = majorVer;

\*minorVersion = minorVer;

/\* Set the TOC buffer **size** to be used to read this container...

2 5

if (majorVer > 1)

\*tocBufSize = ((unsigned Long)bufSize) \* kCMBufSizeUnits...

...called repeatedly for each handler operation type. The operationType is passed to the metahandler to get the **corresponding** handler routine address. if there is one, it is passed back in the handlerAddress parameter and false...

...buffer pointer or a pointer to NULL. It indicates whether the handler for the operationType is a **required** handler or optional. The NULL means it's optional. If not NULL it means the handler is **required** . If the metahandler returns NULL, it means there is no handler for the operationType. Then depending on...missing. After aLt\*/ /\* the handlers have been determined, the missing string will have a List of aLL\*/ /\* **required** handlers that were missing. It can then be used as an error insert.\*/ is if OhandterAddress 11...

...allocation failure...

failure = true; /\* ... everything fails from then on!\*/

else ( /\* if it's not ok and we **need** this handler\*/

2 0 if (\*missing == (char \*)0xFFFFFFFF) /\* if this is the first missing routine... if Wmissing...data in the Container Manager! Enough said. 0 /\* Oho by the way. I know you don't **need** the extra braces in the above init list.\*/ /\* But my MPW C compiler choked without them when...of our standard macros like error\*/ /\* reporting and memory management.

3 0

if ((container = (ContainerPtr)Session#4attoc( **sizeof** (Container))) = NULL)

SessionERROR(C@.err.NoContainer);

return (NULL);

3 5

container->sessionData = sessionData; /\* whew!

container->tocFulltyReadln = false...

...then validate the refNum's container pointer by checking it against this pointer.\*/ 5 0 /\* If it **matches** we're "happy". If not, we assume we don't have a garbage refNum,\*/ /\* This is NOT...

...container);

return (NULL);

0

/\* Build the handler vector. Any missing routines will result in failure with the\*/ /\* **corresponding** interface type names accumulated in the "missing" string which we\*/ /\* use as an error insert. If NULL...

```

...macros. Their names are self-explanatory.
missing (char *)0xFFFFFFFF; /* allocate on first use
0 optional NULL;
#define Required Missing
#define RequiredForWriting ((useFlags & (kCMWriting
kCMReuseFreeSpace))!=0 ? &missing
: &optional)
#define RequiredForReading ((useFlags & (kCMWriting
kCMReuseFreeSpace))==0 ? &missing
5 : &optional)
#define RequiredForUpdating ((useFlags & (kCMUpdateTarget 11
kCMUpdateByAppend))!=0 ?
&missing : &optional)
#define Optional &optional
0 container->handier.cmreturnContainerName = NULL; /* optional, but make
sure we know it!*/ failure = buildHandlerVector(container, (CMHandlerAddr
*)&container->handter.cmfdopen,
CHOpenOpType, Required );
failure 1= buildHandlerVector(container, (CMHandlerAddr
*)&container->handter.cmfclose,
5 CMCloseOpType, Required );
failure 1= buildHandlerVector(container, (CMHandlerAddr
*)&container->handter.cmfflush,
CMFlushOpType, Optional);
failure 1= buildHandlerVector(container, (CMHandlerAddr
*)&container->handter.cmfseek,
CMSeekOpType, Required );
0 failure 1= buildHandlerVector(container, (CMHandlerAddr
*)&container->handter.cmftell,
CMTellOpType, Required );
failure 1= buildHandlerVector(container, (CMHandlerAddr
*)&container->handter.cmfread,
CMReadOpType, RequiredForReading );
failure 1= buildHandlerVector(container, (CMHandlerAddr
*)&container->handter.cmfwrite,
CMWriteOpType, RequiredForWriting );
failure 1= buildHandlerVector(container, (CMHandlerAddr
*)&container->handter.cmfeof,
CMEofOpType, Optional);
failure 1= buildHandlerVector(container, (CMHandlerAddr
*)&container->handter.cmftrunc,
CMTruncOpType, Optional);
0 failure 1= buildHandlerVector(container, (CMHandlerAddr
*)&container->handter.cmgetContainerSize, CMSizeOpType, Required );
failure 1= buildHandlerVector(container, (CMHandlerAddr
*)&container->handier.emreadLabel, CMReadLbLOpType, RequiredForReading
);
failure 1= buildHandlerVector(container, (CMHandlerAddr
*)&container->handier.cmwriteLabel, CMWriteLbLOpType, RequiredForWriting
);
failure 1= buildHandlerVector(container, (CMHandlerAddr
*)&container->handter.cmreturnParentValue, CMParentOpType, Optional);
failure 1= buildHandlerVector(container, (CMHandlerAddr
*)&container->handier...

...buildHandlerVector(container, (CMHandlerAddr
*)&container->Phandler.cmreturnTargetType, CMTargetTypeOpType, Optional);
failure 1= buildHandlerVector(container, (CMHandlerAddr
*)&container->handter.cmextractData, CMExtractDataOpType, Required );
failure 1= buildHandlerVector(container, (CMHandlerAddr
*)&container->handter.cmformatData, CMFormatDataOpType, Required );
if (failure)
if (missing 1= (char *)0xFFFFFFFF && missing != NULL)
ERRORI(CM-err-UndefRoutine, missing);
CMfree(missing);
CMfree...

```



```

...TOC we're using is not for the current container, but for the*/ /*
target container. Thus we need two TOC pointers; one that we usually
use for */ /* accessing the "proper" TOC, and one that always...

...NoTOC, CONTAINERNAME);
Wree(container);
return (NULL);
container->toc = container->privateTOC; /* at this point both TOC ptrs
are equal */ 2 0 container->usingTargetTOC = false; /* ... and there is
no target TOC*/ /* Set up the global name symbol table in much the same
way we did the TOC. We need */ /* two tables here too. 2 5 if
((container->privateGlobals = cmCreateGlobalNameTbl(container)) == NULL)
ERROR1(CM
err.NoGlobalTable. CONTAINERNAME...deletesValueHdr = NULL;
5 container->touchedchain = NULL;
container->iobuffer = NULL;
container->tocIOctl = NULL;
1 0 container->trackFreeSpace = true; /* free space normally always
tracked*/
cmInitList(&container->deletedValues);
cmInitList(&container->embeddedContainers);
cmInitList(&container->activeIOBuffers);
15 /* Determine whether we're...

...we open the file for 2 0 /* appending. /* The modes of container (file)
opening we use here need some discussion. We have
/* three open modes to go with the three open cases:
2 5 converting...

...intent is to create the container if*/ 3 0 it doesn't already exist, set
its file size to 0 (truncate it), and*/
to allow BOTH reading and writing (update). The API allows reading*/
of...

...11. An existing container is to be opened*/
0 for updating. This is also used for reusing free space . */
/* The 'lb' in these modes is just to indicate a binary file is intended
as opposed to...CMfopen(container, attributes, "wb+"); /* ... open update
& trunc else /* if reading... if ((useFlags & kMeuseFreeSpace) != 0) f /*
if reuse free space ... container->refCon =
CMfopen(container, attrixjtes.lrb+");/* ... open for updating*/
container->useFlags = (unsigned short)(useFlags | kCMWriting);
5...

...container->physicatEOF; /* Last byte offset + 1*/
container->tocOffset = tocOffset; /* offset to TOC*/
container->tocSize = tocsiz; /* TOC total size */
0 container->logicaLEOF = tocOffset; /* doesn't include TOC*/
#if CMDUM

PTOC /* show TOC in container*/
if (SESSION...5
The container corresponding to the specified typeName is opened for input
or for updating by reusing free space . The association between the
typeName and the physical container is through the metahandler defined
for that same...

...done. If kCMReuseFreeSpace is specified, than BOTH reading and writing
may be done to update the container. Free space from deleted data
will be reused and overwrites of existing data may be done to change it
(subject to the 5 5 container Label flags, see below). Free space is
always kept track of on a List. it takes the form of standard TOC
entries for TOC ID 1. property COLStdObjID TOC Free. Only space greater
than a TOC entry size is remembered since each free List entry 'cost
at least a TOC entry itself.
6 0...

```

```

...be a pathname. For an embedded container, it probably would be the
parent value 6 5 (Malue), corresponding to the embedded container. When
the open handler is called to open the container, it is given...reading,
pure writing, or updating).*/ /* From here it will always be pure reading
or updating by reusing free space . Note /* that we did not set the
kCMWriting useFlag for updating. it will, however, be set /* once...
...container control block so that we */ /* don't have to hunt it down
again. We don't need it for just reading a container,*/ /* but since we
got the pointer we might as well make...UndoOpen;
return (NULL);
5 0 theVatueHdr->vaLueFLags l= VaLueProtected; /* don't allow writing to
this vaLue*/ /* The size property is used to get at the size in the
TOC object. This mist be in /* agreement with the TOC size in the Label
which we already extracted to Load the /* TOC in the first pLace.Note, we
"Lie" about the size in the TOCvaLueHdr for this*/ /* entry in that the
size we place there is the TOC size and NOT the size of the */ /*
value itself. This allows CMGetVaLueSize0 to operate without any special
cases.*/
theProperty = cmGetObjectProperty(theTOCObject, CH...

...err BadSize, CONTAINERNAME);
UndoOpen(& @
return (NULL);
theVaLueHdr->vatuueFLags VaLueProtected; /* don't allow writing to this
value*/
theVaLueNdr-> size container->tocSize; /* "Lie" about the size
0 container->tocobjValue theSizeVaLue; /* save ptr to size value for
updating*/ /* For updates, there is a TOC object value which represents
the entire container, /* from first byte (always offset 0) to end of
Label. For in-pLace updates, the */ /* container size could change so
this value's size must be adjusted accordingly. 1 5 /* For appended
updates we are Layering additional TOCs. The value in track of the total
amount of free space . theProperty = cmGetObjectProperty(theTOCObject,
CM-StdObjID.TOC.Deleted);
4 0 if (theProperty) (
theValueHdr = (TOCvaLueHdrPtr)cmGetListHead(&theProperty->vaLueHdrList);
if...

...CONTAINERNAME);
5 UndoOpen(T;
return (NULL);
theVaLueHdr->vaLueFLags l= VatuueProtected; /* don't allow writing
1 0
/* The "free space" property contains a single value header with
value segments /* defining the free space list. There may not be any
"free space" property if there*/ /* is no free space . If there
is, we save the pointer to the value header in the*/ /* container to make
it more efficient to maintain the free space List. If there is*/ /*
none, the pointer remains NULL. The cmRememberFreeDataSpaceo routine will
create*/ /* the property the first time it is called to record free
space .
theProperty = cmGetObjectProperty(theTOCObject, CM.StdObjID.TOC-Free);
if (theProperty) (
0 container->freeSpaceVaLueHdr
(TOCvaLueHdrPtr)cmGetListHead(&theProperty->vaLueHdrList);
if...reuse updating, then we take the container space used /* by the TOC
and add it to the free space List. In standard ANSI C 1/0, which we
/* assume the 1/0 handlers can be written...if not dynamic value type...
(targetType = CMreturnTargetType(container)) == NULL) (
ERROR1(CM err NoTypeHandler, CONTAINERNAME); /* ... error (it's required
here)*/
UndoOpen; ,
return (NULL);
5 0
/* The handler did indeed return a type. Now we will attempt...

```

...updates to /\* be applied to a yet "deeper" target. To be able to do

```

this, ALL targets need to 0 /* be able to get at the top-most container
(the one we're opening here...one we opened for converting,*/ 0 /* the
value data for the value is the entire container's Limits , i.e., offset
0 with /* a size equal to the current physical EOF.
targetProperty = CMRegisterProperty((CMContainer)container,
(CMGlobalName)CMTOCTargetGlobalName);
if (targetProperty != NULL) {
targetType = CMRegisterType((CMContainer)...kCMReuseFreeSpace may also be
specified in conjunction with the others just mentioned. Here it only
means that free space will try to be reused when possible. Unlike
MpenContainer, you always can read and write, so it...

...will be discussed shortly. updating here is indicated by
kCMUpdateByAppend or kCMUpdateTarget to have a special meaning. Free
space is always kept track of on a List. It takes the form of
standard TOC
5...

...entries for TOC ID 1. property CM
StdObjID TOC Free. Only space greater than a TOC entry
size is remembered since each free list entry costs at least a TOC
entry itself. With...business we can think of in the future). 2 5
cmSetValueBytes(container, &valueBytes, Value.Lmm.Long, MinUserObjectID,
sizeof (CM ULONG));
theTOCObject = (CObject)cmDefineObject(container, CM.StdObjID.TOC,
CM.StdObjID.TOC.Seed,
CM.StdObjID.TOC.Type...
...valueFlags |= ValueProtected; /* don't allow writing to this value*/ /*
The minimum seed value is always initially equal to the seed value. It
is used /* when applying updates to a target container. For a newly...

...what the seed*/ /* should be (a "chicken-and-egg" problem).
cmSetValueBytes(container, &valueBytes, Value.Lmm.Long, MinUserObjectID,
sizeof (OLULONO);
5 0 theTOCObject = (CObject)cmDefineObject(container, CM.StdObjID.TOC,
CM.StdObjID.TOC.MinSeed,
CM,...StdObjID.TOC...

...to this value*/ /* To allow the user to treat the entire TOC itself as
an object, we need to define a /* property of the TOC object with the
offset and size . it is used as an additional /* check for reading
containers. More importantly we need this to allow CMUseValue0 /* and
CMGetValueSize0 to operate. CMGetValueSize0 will always return 0 while /*
/* writing, but the actual size when reading. The bad part is we have a
"chicken and /* egg" problem setting the value. We...

...TOC object 1 has been written to the container. This*/ /* means we have
to back patch the size in the container. Thus we use the pointer to /*
the value as a signal to remember where...
...the entire container as a single object (from 1st byte to the*/ /* end
of the Label), we need to define a property of the TOC object with an
offset /* /* equal to the first available byte (0), and a size which
cannot be determined until *1 /* we write the entire TOC (so we save a
pointer to...

...in conjunction with the total /* deleted space value (defined below) to
get a fragmentation estimate. However, we /* need it for
update-by-append edits so that an editing container can "get at" the 2 0
...

...when we have layered TOCs. Note, that the offset for this*/ /* value
will always be 0. The size changes due to initial container creation
being*/ /* initiated here, or by updates initiated by CMOpenContainer.
cmSetValueBytes(container...

...That*/
/* value is created here and now... *1

```

```
3 5 cmSetVaLueBytes(container, &vaLueBytes, Value-Imm Long, 0, sizeof
(CM ULONG)); cmDefineObject(container, CM StdObj1D TOC, CM Std6bjID TOC
Deleted,
CM
StdObj1D - TOC Type, lvalueByies, gene...
```

16/5,K/19 (Item 7 from file: 349)  
DIALOG(R)File 349:PCT FULLTEXT  
(c) 2003 WIPO/Univentio. All rts. reserv.

00281629

**WRITE ANYWHERE FILE-SYSTEM LAYOUT**

**DISPOSITION D'UN SYSTEME DE FICHIERS A ECRITURE DANS UNE ZONE NON  
PREDETERMINEE**

Patent Applicant/Assignee:

NETWORK APPLIANCE CORPORATION,

Inventor(s):

HITZ David,

MALCOM Michael,

LAU James,

RAKITZIS Byron,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9429807 A1 19941222

Application: WO 94US6320 19940602 (PCT/WO US9406320)

Priority Application: US 93643 19930603

Designated States: JP AT BE CH DE DK ES FR GB GR IE IT LU MC NL PT SE

Main International Patent Class: **G06F-015/40**

Publication Language: English

Fulltext Availability:

Detailed Description

Claims

Fulltext Word Count: 15104

**English Abstract**

The present invention provides a method for keeping a file system in a consistent state and for creating read-only copies of a file system. Changes to the file system are tightly controlled. The file system progresses from one consistent state to another. The set of self-consistent blocks on disk that is rooted by the root inode is referred to as a consistency point. To implement consistency points, new data is written to unallocated blocks on disk. A new consistency point occurs when the fsinfo block (2440) is updated by writing a new root inode for the inode file (1210) into it. Thus, as long as the root inode is not updated, the state of the file system represented on disk does not change. The present invention also creates snapshots (Figure 22) that are read-only copies of the file system. A snapshot uses no disk space when it is initially created. It is designed so that many different snapshots can be created for the same file system. Unlike prior art file systems that create a clone by duplicating the entire inode file and all of the indirect blocks, the present invention duplicates only the inode that describes the inode file. A multi-bit free-block map file (1630) is used to prevent data from being overwritten on disk.

**French Abstract**

Procede destine a maintenir un systeme de fichiers dans un etat coherent et a creer des copies de lecture seule d'un systeme de fichiers. Les modifications apportees au systeme de fichiers sont etroitement commande'es. Le systeme de fichiers progresse d'un etat coherent a un autre. La serie de blocs auto-coherents sur disque qui est ancree par l'inode racine est designee sous le terme de point de coherence. Pour mettre en oeuvre des points de coherence, des nouvelles donnees sont ecrites sur des blocs non attribues du disque. Un nouveau point de coherence a lieu lorsque le bloc fsinfo (2440) est mis a jour grace a l'ecriture d'un nouvel inode racine pour le fichier inode (1210) se trouvant dans ledit bloc. Par consequent, tant que l'inode racine n'est pas mis a jour, l'etat du systeme de fichiers represente sur le disque ne change pas. La presente invention concerne egalement des programmes

d'analyse selective (figure 22) qui sont des copies a lecture seule du systeme de fichiers. Un programme d'analyse selective n'utilise pas d'espace sur le disque quand il est initialement cree. Il est concu de telle maniere que de nombreux programmes d'analyse selective differents puissent etre crees pour le meme systeme de fichiers. Au contraire des systemes de fichier de la technique anterieure qui creent un clone en copiant seulement le fichier inode integral et tous les blocs indirects, la presente invention copie seulement l'inode qui decrit le fichier inode. Un fichier (1630) topographique a bits multiples et blocs libres est utilise pour empecher des donnees d'etres ecrasees sur le disque.

Main International Patent Class: **G06F-015/40**

Fulltext Availability:

Detailed Description

Detailed Description

... entry 1110A-1110C for each 4 KB block in the disk system. It also serves as a **free - block** map file. The blkmap file 1110 indicates whether or not a disk block has been allocated. Figure...

...1110A is comprised of 32 bits (BIT0-BIT31). Bit 0 (BIT0) of entry 1110A is the active **file system** bit (FS-BIT). The FS-bit of entry 1110A indicates whether or not the corresponding block is part of the active **file system**. Bits 1-20 (BIT1-BIT20) of entry 1110A are bits that indicate ...block is part of a corresponding - 22 snapshot 1 The next upper 10 bits (BIT21-BIT30) are **reserved**. Bit 31 (BIT31) is the consistency point bit (CP-BIT) of entry 1110A.

A block is available...

16/5,K/20 (Item 8 from file: 349)

DIALOG(R)File 349:PCT FULLTEXT

(c) 2003 WIPO/Univentio. All rts. reserv.

00239212

**METHOD FOR IMPROVING PARTIAL STRIPE WRITE PERFORMANCE IN DISK ARRAY SUBSYSTEMS**

**PROCEDE D'AMELIORATION DES PERFORMANCES D'ECRITURE SUR BANDE PARTIELLE DANS DES SOUS-SYSTEMES A PILE DE DISQUES**

Patent Applicant/Assignee:

COMPAQ COMPUTER CORPORATION,

Inventor(s):

NEUFELD E David,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9313478 A1 19930708

Application: WO 92US10953 19921218 (PCT/WO US9210953)

Priority Application: US 91118 19911227

Designated States: AT AU BG BR CA CH CS DE DK ES FI GB HU JP KR NL NO PL RO

RU SE AT BE CH DE DK ES FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM

GA GN ML MR SN TD TG

Main International Patent Class: **G06F-011/10**

Publication Language: English

Fulltext Availability:

Detailed Description

Claims

Fulltext Word Count: 8742

English Abstract

A method and apparatus for improving disk performance during partial stripe write operations in a computer system having a disk array subsystem utilizing parity fault tolerance technique. When a partial stripe write generation is begun, the method determines if the area or stripe where the write is to occur is unused space in the file system. If not, the partial stripe write operation is performed using a preceding read operation to read the current data and parity information from the

disk as would normally be done. However, if the write area is unused space in the file system, then the contents of the data stripe do not need to be preserved. In this instance, the partial stripe write operation can be performed without any preceding read operations. By obviating the necessity of a preceding read operation, much of the performance penalty of doing a partial stripe write in the case where the rest of the data stripe does not need to be preserved is removed.

#### French Abstract

Procédé et appareil permettant d'améliorer les performances d'un disque au cours d'opérations d'écriture sur bande partielle dans un système informatique comprenant un sous-système à pile de disques, à l'aide d'une technique d'insensibilisation aux défaillances par parité. Lorsqu'une opération d'écriture sur bande partielle est démarrée, le procédé permet de déterminer si la région ou la bande où l'opération d'écriture doit être effectuée constitue un espace inutilisé dans le système de fichiers. Si ce n'est pas le cas, l'opération d'écriture sur bande partielle est effectuée avec une opération de lecture antérieure afin de lire les informations de parité et des données courantes sur le disque comme il est courant de le faire. Cependant, si la zone d'écriture constitue un espace inutilisé dans le système de fichiers, le contenu de la bande de données ne doit pas nécessairement être préservé. Dans ce cas, l'opération d'écriture sur bande partielle peut être effectuée sans qu'il soit nécessaire d'effectuer des opérations de lecture antérieures. Dans la mesure où la nécessité d'effectuer une opération de lecture antérieure est supprimée, une partie importante des pertes de performances encourues au cours d'une opération d'écriture sur bande partielle, dans le cas où il n'est pas nécessaire de préserver le reste de la bande de données, est annulée.

Main International Patent Class: G06F-011/10

Fulltext Availability:

Detailed Description

Detailed Description

... fault tolerant disk array.

Background on file systems used in computer systems is deemed appropriate. Generally, a **file system** may use one of two techniques,, either a "free list" or a bit map technique to describe the amount and location of **free space** on disk drive units. In a free list technique, a known location on the disk contains a...

...a list of other free blocks, i.e. blocks on the disk that are unused by the **file system**. The very last pointer in this block points to a block with a list of other free blocks, thus forming a chain of blocks which contain information about **free space** in the **file system**. When a free list technique is used, it is difficult to determine if a respective block is...

...if the block is free. In a bit map scheme, a portion of the disk includes a **reserved** area where one bit is allocated for every "allocation cluster,," wherein an allocation cluster may include a...

16/5,K/21 (Item 9 from file: 349)  
DIALOG(R) File 349:PCT FULLTEXT  
(c) 2003 WIPO/Univentio. All rts. reserv.

00239209 \*\*Image available\*\*

METHOD FOR PERFORMING DISK ARRAY OPERATIONS USING A NONUNIFORM STRIPE SIZE MAPPING SCHEME

PROCEDE PERMETTANT D'EFFECTUER DES OPERATIONS SUR DES ENSEMBLES DE DISQUES EN UTILISANT UN SYSTEME DE TOPOGRAPHIE A TAILLE DE BANDE NON UNIFORME

Patent Applicant/Assignee:

COMPAQ COMPUTER CORPORATION,

Inventor(s):

NEUFELD E David,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9313475 A1 19930708

Application: WO 92US11283 19921218 (PCT/WO US9211283)

Priority Application: US 91 19911227

Designated States: AT AU BG BR CA CH CS DE DK ES FI GB HU JP KR NL NO PL RO

RU SE AT BE CH DE DK ES FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM

GA GN ML MR SN TD TG

Main International Patent Class: G06F-003/06

International Patent Class: G11B-20:18; G06F-11:10

Publication Language: English

Fulltext Availability:

Detailed Description

Claims

Fulltext Word Count: 9226

#### English Abstract

A method and apparatus for improving disk performance in a disk array subsystem. A nonuniform mapping scheme is used wherein the disk array includes regions having varying sizes of data stripes. The disk array includes a region comprised of data stripes having a stripe size that corresponds to the size of the internal data structures frequently used by the file system, in addition to a region comprised of a number of data stripes having a larger stripe size which are used for general data storage. When a write operation occurs involving one of the data structures, the data structure is preferably mapped to the small stripe region in the disk array having a size which matches the size of the data structure. In this manner, whenever a file system data structure is updated, the operation is a full stripe write. This removes the performance penalty associated with partial stripe write operations.

#### French Abstract

Procédé et appareil d'amélioration de la capacité d'un disque dans un sous-système d'un ensemble de disques. Un système de topographie non uniforme est utilisé selon lequel l'ensemble de disques comprend des régions ayant des tailles variables de bandes de données. La pile de disques comprend une région constituée de bandes de données ayant une dimension de bande qui correspond à la dimension des structures de données internes fréquemment utilisées par le système de fichier, en plus d'une région constituée de plusieurs bandes de données ayant une dimension de bande plus grande que celles qui sont utilisées pour le stockage général de données. Lorsqu'une opération d'écriture impliquant l'une des structures des données a lieu, la structure de données est de préférence topographiée sur la petite région de bande dans la pile de disques ayant une dimension qui correspond à la dimension de la structure de données. De cette manière, à chaque fois qu'une structure de données du système de fichier est mise à jour, l'opération est une écriture sur une bande pleine. Ceci élimine le désavantage d'opérations d'écriture sur bande partielle.

Main International Patent Class: G06F-003/06

...International Patent Class: G06F-11:10

Fulltext Availability:

Detailed Description

#### Detailed Description

... As described in the background, the UNIX operating system includes a service referred to as the make file system program. In the preferred embodiment, the make file system program provides information to the disk controller 112 as to how many INODEs are being created and the size of the INODEs, Optionally, the make file system includes sufficient intelligence to inform the disk controller 112 as to the desired-stripe

size in the...

...to be allowed in the system, The disk array controller 112 uses this information to develop the **file system** on each of the disks comprising the array 116, The disk array controller 112 uses a multiple...

...stripe regions, The small stripe region preferably occupies the first N sectors of each disk and is **reserved** for the INODE data structures, and the remaining stripes in the array form the large stripe region, which comprises **free space** used for data storage, Therefore, in the preferred embodiment, the disk controller 112 allocates the first N...

16/5,K/23 (Item 11 from file: 349)  
DIALOG(R) File 349:PCT FULLTEXT  
(c) 2003 WIPO/Univentio. All rts. reserv.

00153060

**PARALLEL MACHINE ARCHITECTURE FOR PRODUCTION RULE SYSTEMS**  
**ARCHITECTURE DE MACHINE PARALLELE POUR DES SYSTEMES DE REGLES DE PRODUCTION**  
Patent Applicant/Assignee:

MARTIN MARIETTA ENERGY SYSTEMS INC,

Inventor(s):

ALLEN John Daniel Jr,

BUTLER Philip Lee,

Patent and Priority Information (Country, Number, Date):

Patent: WO 8809972 A1 19881215

Application: WO 88US1901 19880609 (PCT/WO US8801901)

Priority Application: US 87976 19870609

Designated States: AT BE CH DE FR GB IT JP LU NL SE

Main International Patent Class: **G06F-015/18**

Publication Language: English

Fulltext Availability:

Detailed Description

Claims

Fulltext Word Count: 138162

English Abstract

A parallel processing system (2) for production rule programs utilizes a host processor (4) for storing production rule right hand sides (RHS) and a plurality of rule processors (6) for storing left hand sides (LHS). The rule processors operate in parallel in the recognize phase of the system recognize -Act Cycle- to match their respective LHS's against a stored list of working memory elements (WME) in order to find a self-consistent set of WME's. The list of WME is dynamically varied during the Act phase of the system in which the host executes or fires rule RHS's for those rules for which a self-consistent set has been found by the rule processors. The host (4) transmits instructions for creating or deleting working memory elements as dictated by the rule firings until the rule processors are unable to find any further self-consistent working memory element sets at which time the production rule system is halted.

French Abstract

Un systeme de traitement en parallele (2) pour la production de programmes de regles utilise un ordinateur central (4) pour stocker des parties droites (RHS) de regles de production et une pluralite de processeurs de regles (6) pour stocker des parties gauches (LHS). Les processeurs de regles fonctionnent en parallele dans la phase de reconnaissance du systeme - cycle d'action - pour faire correspondre leurs LHS respectives avec une liste memorisee d'elements memoire de travail (WME) de maniere a trouver un ensemble autoconsistant d'elements memoire de travail (WME). La liste des WME varie dynamiquement pendant la phase d'action du systeme dans lequel l'ordinateur central execute ou



declenche les RHS pour les regles dont un ensemble autoconsistant a ete trouve par les processeurs de regles. L'ordinateur central (4) transmet des instructions pour creer ou effacer des elements memoire de travail comme cela est dicte par les declenchements de regle jusqu'a ce que les processeurs soient invalides dans le but de trouver d'autres eventuels ensembles d'elements memoire de travail autoconsistants, moments pendant lesquels le systeme de regle de production est arrete.

Main International Patent Class: **G06F-015/18**

Fulltext Availability:

Detailed Description

Detailed Description

... the square JEDEC 68000 packages and 256K x 8 single-in-line hybrid memory modules are most **space** efficient. For control logic, a PLA and latch could form a control state machine for memory timing...

...microprocessors are placed on each of four Multibus boards for a total of 16 rule processors.

The **system** may, of course, be expanded to include 128 rule processors or even a larger number if desired...on the interface board.

I-WINDOW\*

Window Select Signal. The cell that is selected by the window **number** on the window number bus is actually selected when this line is low.

I-STATEN\*

Board Status...critical, so the search time to add a new string to the token package is not a **limiting** factor.

S U E Z S T I I wirlj

The next paragraphs will show a general sequence of operations **required** for the parallel processing OPS initialization and program compilation. When OPS is first started from CP/M...new rule, other algorithms may be used, but the above algorithm is advantageous since it does not **require** a look ahead approach, The OPS parser is a one pass compiler, As each rule is parsed...Both kinds of frames can be resized or removed at any time, however the static frames are **reserved** for the type of data that does not require very much resizing. Therefore, static frames are kept...

16/5,K/24 (Item 12 from file: 349)

DIALOG(R) File 349:PCT FULLTEXT

(c) 2003 WIPO/Univentio. All rts. reserv.

00106554 \*\*Image available\*\*

**DATA PROCESSING SYSTEM**

**SYSTEME DE TRAITEMENT DE DONNEES**

Patent Applicant/Assignee:

INTEL CORP,

Inventor(s):

COLLEY S,  
RATTNER J,  
COX G,  
SWANSON R,

Patent and Priority Information (Country, Number, Date):

Patent: WO 8102477 A1 19810903

partitions including total files and directories, cluster size, used space, **free space**, wasted space, and **file system** parameters through multiple virtual operations; and undo operations before committing the changes.

One embodiment of the invention...

15/5,K/3 (Item 3 from file: 348)  
DIALOG(R) File 348:EUROPEAN PATENTS  
(c) 2003 European Patent Office. All rts. reserv.

00831802

**File system**

**Dateiensystem**

**Système de fichiers**

PATENT ASSIGNEE:

MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD., (216883), 1006, Oaza Kadoma, Kadoma-shi, Osaka-fu, 571, (JP), (applicant designated states: DE;FR;GB)

INVENTOR:

Karibe, Tomoyuki, 3-14-110, Miyukihigashimachi, Neyagawa-shi, Osaka-fu, (JP)

Kokado, Takeshi, 27-1-207, Nishikyuhoji-cho, Okamedani, Fukakusa, Fushimi-ku, Kyoto-shi, Osaka-fu, (JP)

Ito, Yukiko, 6-Nishi 2-320, Sotojima-cho, Moriguchi-shi, Osaka-fu, (JP)

Tanaka, Tsutomu, 10-5-38, Uegahara, Nishinomiya-shi, Hyogo-ken, (JP)

Tamai, Masaaki, 2-26-3, Yakumokitamachi, Moriguchi-shi, Osaka-fu, (JP)

Doi, Shinzo, 2-4-32, Makinohonmachi, Hirakata-shi, Osaka-fu, (JP)

LEGAL REPRESENTATIVE:

Altenburg, Udo, Dipl.-Phys. et al (1268), Patent- und Rechtsanwälte, Bardehle . Pagenberg . Dost . Altenburg . Frohwitter . Geissler & Partner, Galileiplatz 1, 81679 Munchen, (DE)

PATENT (CC, No, Kind, Date): EP 770964 A1 970502 (Basic)

APPLICATION (CC, No, Date): EP 96117157 961025;

PRIORITY (CC, No, Date): JP 95278799 951026; JP 95278813 951026; JP 95278814 951026; JP 96125146 960520; JP 96213556 960813

DESIGNATED STATES: DE; FR; GB

INTERNATIONAL PATENT CLASS: **G06F-017/30**

ABSTRACT EP 770964 A1

When a get resource message arrives from an exterior, a resource allocation part 102 reserves a resource. In case of data reading or data writing, a declared value management part 111 checks parameters and a slot allocation part 103 allocates the resource, thereby constructing an efficient file system guaranteeing delay quality in relation to data reading. As the result, it is possible to provide a file system providing pictures in a quick response time for requests from users and supporting users requiring various regeneration speeds.

ABSTRACT WORD COUNT: 86

LEGAL STATUS (Type, Pub Date, Kind, Text):

Application: 970502 A1 Published application (A1with Search Report ;A2without Search Report)

Examination: 971105 A1 Date of filing of request for examination: 970909

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text Language Update Word Count

CLAIMS A (English) EPAB97 7899

SPEC A (English) EPAB97 39029

Total word count - document A 46928

Total word count - document B 0

Total word count - documents A + B 46928

INTERNATIONAL PATENT CLASS: **G06F-017/30**

...CLAIMS block read request being not inputted in said exchange unit among

said block read requests.

19. The **file system** according to claim 16, further comprising a resource management part managing a maximum value of read or...

...at said interval T(underscore)min when a band necessary for reading a file is ensured;

to **reserve** said expected slot if a **free space** is present in said decided expected slot;

to successively refer to an MS arrival slot of a next time for **reserving** an MS arrival slot having a **free space** if said decided expected slot has no **free space**; and

to allocate said block read request to said **reserved** MS arrival slot in case of data reading.

20. The file system according to claim 19, wherein...

15/5,K/4 (Item 4 from file: 348)

DIALOG(R)File 348:EUROPEAN PATENTS

(c) 2003 European Patent Office. All rts. reserv.

00306062

Digital data processing system.

Digitales Datenverarbeitungssystem.

Systeme du traitement de donnees numeriques.

PATENT ASSIGNEE:

DATA GENERAL CORPORATION, (410940), Route 9, Westboro Massachusetts 01581  
, (US), (applicant designated states: AT;BE;CH;DE;FR;GB;IT;LI;LU;NL;SE)

INVENTOR:

Bratt, Richard Glenn, 9 Brook Trail Road, Wayland Massachusetts 01778,  
(US)

Clancy, Gerald F., 13069 Jaccaranda Center, Saratoga California 95070,  
(US)

Gavrin, Edward S., Beaver Pond Road RFD 4, Lincoln Massachusetts 01773,  
(US)

Gruner, Ronald Hans, 112 Dublin Wood Drive, Cary North Carolina 27514,  
(US)

Mundie, Craig James, 136 Castlewood Drive, Cary North Carolina, (US)

Schleimer, Stephen I., 1208 Ellen Place, Chapel Hill North Carolina 27514  
, (US)

Wallach, Steven J., 12436 Green Meadow Lane, Saratoga California 95070,  
(US)

LEGAL REPRESENTATIVE:

Robson, Aidan John et al (69471), Reddie & Grose 16 Theobalds Road,  
London WC1X 8PL, (GB)

PATENT (CC, No, Kind, Date): EP 300516 A2 890125 (Basic)

EP 300516 A3 890426

EP 300516 B1 931124

APPLICATION (CC, No, Date): EP 88200921 820521;

PRIORITY (CC, No, Date): US 266413 810522; US 266539 810522; US 266521

810522; US 266415 810522; US 266409 810522; US 266424 810522; US 266421

810522; US 266404 810522; US 266414 810522; US 266532 810522; US 266403

810522; US 266408 810522; US 266401 810522; US 266524 810522

DESIGNATED STATES: AT; BE; CH; DE; FR; GB; IT; LI; LU; NL; SE

RELATED PARENT NUMBER(S) - PN (AN):

EP 67556 (EP 823025960)

INTERNATIONAL PATENT CLASS: G06F-009/46 ; G06F-012/14

CITED REFERENCES (EP A):

PROCEEDINGS OF THE SPRING JOINT COMPUTER CONFERENCE, Atlantic City, 1972,  
pages 417-429, Afips Press; G.S. GRAHAM et al.: "Protection-Principles  
and practice"

IDEM.

COMPCON SPRING'80, digest of papers, San Francisco, 25th-28th February  
1980, pages 340-343, IEEE, New York, US; T.D. McCREERY: "The X-tree  
operating system: Bottom layer"

gere une liste resultante (506) d'operations de manipulation de partition, liste susceptible d'etre optimisee (512). Cette liste peut egalement etre realisee automatiquement par un moteur classique (210) de manipulation de partition sans exiger d'entree utilisateur (100) supplementaire a la fin de chaque entree de liste (212). Il est possible, grace a cette invention, de manipuler automatiquement des partitions etendues (804) et de se procurer une assistance s'agissant de tele-manipulation de partition par le biais d'une architecture d'interface utilisateur a deux parties.

Main International Patent Class: **G06F-012/00**

Fulltext Availability:

Detailed Description

Detailed Description

... intervention- retain physical attributes of the virtual partitions including total files and directories, cluster size, used space, **free space**, wasted space, and **file system** parameters through multiple virtual operations; and undo operations before committing the changes.

One embodiment of the invention...

**15/5,K/13 (Item 7 from file: 349)**

DIALOG(R) File 349:PCT FULLTEXT

(c) 2003 WIPO/Univentio. All rts. reserv.

00484640 \*\*Image available\*\*

**DATA PROCESSING SYSTEM**

**SYSTEME DE TRAITEMENT DE DONNEES**

Patent Applicant/Assignee:

BRITISH TELECOMMUNICATIONS PUBLIC LIMITED COMPANY,  
FOWLER Jonathan Stephen,

Inventor(s):

FOWLER Jonathan Stephen,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9915992 A1 19990401

Application: WO 98GB2845 19980921 (PCT/WO GB9802845)

Priority Application: GB 9720395 19970924

Designated States: AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES

FI GB GE GH GM HR HU ID IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD

MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG US

UZ VN YU ZW GH GM KE LS MW SD SZ UG ZW AM AZ BY KG KZ MD RU TJ TM AT BE

CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN

GW ML MR NE SN TD TG

Main International Patent Class: **G06F-017/30**

Publication Language: English

Fulltext Availability:

Detailed Description

Claims

Fulltext Word Count: 9752

English Abstract

A data processing application processes data files and generates storage operation instructions for data files identified by identification information independent of the storage location of the data files. A plurality of storage locations are provided for storing data files and a storage map stores information on the storage locations of stored data files in the storage locations and identification information for the stored data files. The identification information generated by the data processing application is used to look up a storage location information and a storage interface receives the storage location information and storage operation instructions indicating a storage operation to be carried out and carries out the storage operation instructions for the data file in a storage location indicated by the storage location information.

French Abstract

La presente invention concerne une application informatique traitant des fichiers de donnees et generant des instructions de memoire pour des fichiers de donnees identifies par des informations d'identification independantes de l'emplacement memoire des fichiers de donnees. Une pluralite d'emplacements memoire est fournie de facon a mettre des fichiers de donnees en memoire. Un topogramme memoire conserve, d'une part des informations sur les emplacements memoire des fichiers de donnees mis en memoire dans ces emplacements memoire, et d'autre part des informations d'identification concernant les fichiers de donnees en memoire. Les informations d'identification generees par l'application informatique sont utilisees pour consulter des informations d'emplacement memoire. Une interface, qui recoit les informations d'emplacement memoire et les instructions de memoire indiquant une operation de memoire a effectuer, execute les instructions de memoire concernant le fichier de donnees dans un emplacement memoire indique par les informations d'emplacement memoire.

Main International Patent Class: **G06F-017/30**

Fulltext Availability:

Detailed Description

Detailed Description

... a reference to Figure 1 1 .

Type indicates the type of file which is stored for that **file system** number. The status indicates whether the storage location is available (A), read-only (N) or shut down...

...the physical location assigned for the storage of the files. This can either be simply the directory **reserved** for storage of the files on the local processor, or in a network arrangement this can comprise an identification of the machine or node and the directory on that machine. Columns also indicate the **free space** currently available for that area and the total space available for that area.

5 Once merge processing...

15/5,K/14 (Item 8 from file: 349)

DIALOG(R) File 349:PCT FULLTEXT

(c) 2003 WIPO/Univentio. All rts. reserv.

00376923

**STRUCTURED FOCUSED HYPERTEXT DATA STRUCTURE**

**STRUCTURE DE DONNEES HYPERTEXTE ARTICULEE SUR LA STRUCTURATION**

Patent Applicant/Assignee:

HYPERMED LTD,  
OREN Avraham,  
OLCHA Lev,  
KOWALSKI Nahum,  
MARGULYAN Rita,

Inventor(s):

OREN Avraham,  
OLCHA Lev,  
KOWALSKI Nahum,  
MARGULYAN Rita,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9717666 A2 19970515

Application: WO 9611131 19961023 (PCT/WO IL9600131)

Priority Application: US 95551929 19951023

Designated States: AL AM AT AU AZ BB BG BR BY CA CH CN CZ DE DK EE ES FI GB

GE HU IS JP KE KG KP KR KZ LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL

PT RO RU SD SE SG SI SK TJ TM TR TT UA UG US VZ VN KE LS MW SD SZ UG AM

AZ BY KG KZ MD RU TJ TM AT BE CH DE DK ES FI FR GB GR IE IT LU MC NL PT

SE BF BJ CF CG CI CM GA GN ML MR NE SN TD TG

Main International Patent Class: **G06F-017/30**

International Patent Class: **G06F-17:21**

Publication Language: English  
Fulltext Availability:  
Detailed Description  
Claims  
Fulltext Word Count: 263802

#### English Abstract

A hypertexted data structure (3/16) stored on a computer readable memory device and organized in a hierarchy of at least two levels, the data structure comprising: a plurality of data units (18-20) positioned at different levels in the hierarchy each containing at least some textual information (23) and a plurality of hypertext links (1) each linking at least part of the textual information in a given source data unit to a target data unit; wherein at least one of the hypertext links (1) is linked to at least one hypertext node (34) which contains information relating at least to both the given source data unit and the target data unit from which the relative positions in the hierarchy of the given source and target data units linked by the hypertext link may be determined.

#### French Abstract

La presente invention concerne une structure de donnees en format hypertexte (3/16) stockees dans une memoire lisible par ordinateur et organisee selon une hierarchie comportant au moins deux niveaux. Cette structure de donnees est constituee, d'une part de plusieurs unites de donnees (18-20) se placant a differents niveaux de la hierarchie, chacune de ces unites de donnees contenant au moins quelques donnees textuelles (23), et d'autre part, d'un jeu de liens hypertexte (1), chacun de ces liens reliant au moins une partie de l'information textuelle d'une unite de donnees origine specifique a une unite de donnees cible. L'un au moins des liens hypertexte (1) est relie a l'un au moins des noeuds hypertexte (34) qui contient des donnees se rapportant au moins a la fois a l'unite de donnees origine specifique et a l'unite de donnees cible a partir de laquelle il est possible de determiner des positions relatives dans la hierarchie. Ces positions relatives sont celles des unites de donnees origine et cible reliees par le lien hypertexte.

Main International Patent Class: **G06F-017/30**

International Patent Class: **G06F-17:21**

Fulltext Availability:  
Detailed Description

#### Detailed Description

```
... Direction ' words so that a leading
GetFromBigArray k, sc space is expected
'And if ther is no space ,
just a - or / or other
new code 8/11 'delimiter, this will not
i = InStr(1, sc...names " & INListStart)
for display of the If i > 0 Then
progress of the search to the user ' need to strip it off
RememberSearchString
PtrNameElemStack Left$(RememberSearchString, i - 1)
PtrNameElemStack + I End If
NameElemStack(PtrNameElemStack) I...Search" ' add another Case option
frmMainSearch.ListOfMatches.Clear I
frmMainSearch.ListOfMatches.Addlte Select Case Op
m "No matches found" Case ANDSymbol, ORSymbol
I multiple character operators
End Sub TernpOp = Op
Case Else
Sub SwapDataArrays (ii...
```

00367135      \*\*Image available\*\*

**METHOD AND APPARATUS FOR STRIPING DATA AND FOR ADDING/REMOVING DISKS IN A RAID STORAGE SYSTEM**

**PROCEDE ET APPAREIL POUR REPARTIR DES DONNEES ET POUR AJOUTER OU ENLEVER DES DISQUES DANS UN SYSTEME DE STOCKAGE RAID**

Patent Applicant/Assignee:

BORG TECHNOLOGIES INC,

Inventor(s):

STALLMO David C,

HALL Randy K,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9707462 A1 19970227

Application: WO 96US13423 19960815 (PCT/WO US9613423)

Priority Application: US 95516232 19950817

Designated States: CA AT BE CH DE DK ES FI FR GB GR IE IT LU MC NL PT SE

Main International Patent Class: G06F-012/00

International Patent Class: G06F-12:16 ; G06F-11:10 ; G06F-11:20

Publication Language: English

Fulltext Availability:

Detailed Description

Claims

Fulltext Word Count: 13635

**English Abstract**

Methods for striping and unstriping data on a plurality of storage devices in a redundant array of independant disks (RAID) system, and for adding, removing and organizing storage devices in the RAID system, are disclosed. The method for striping and unstriping data comprises the steps of dividing data blocks on the plurality of storage devices into a plurality of square portions and exchanging the data in the sets of blocks of each of the plurality of square portions. The exchanging step includes the steps of selecting a square portion (1204), locating a diagonal set of blocks within the selected square portion (1212), and exchanging all sets of blocks equidistant from the diagonal set of blocks, on opposite sides of the diagonal set of blocks, and in a line perpendicular to the diagonal set of blocks (1214, 1216, 1218, 1220 and 1222).

**French Abstract**

L'invention porte sur un procede pour repartir et deplacer des donnees sur une pluralite de dispositifs de stockage dans un systeme RAID (pile redondante de disques independants), et pour ajouter, enlever et organiser des dispositifs de stockage dans le systeme RAID. Le procede pour repartir et deplacer les donnees comprend plusieurs etapes: la division des blocs de donnees en une pluralite de parties carrees sur une pluralite de dispositifs de stockage, et l'echange des donnees dans les series de blocs de chacune des parties carrees. L'etape d'echange comprend la selection d'une partie carree (1204), la localisation d'une serie diagonale de blocs dans les limites de la partie carree selectionnee (1212), et l'echange de toutes les series de blocs equidistantes de la serie diagonale de blocs, sur les cotes opposes de la serie diagonale de blocs, et sur une ligne perpendiculaire a la serie diagonale de blocs (1214, 1216, 1218, 1220 et 1222).

Main International Patent Class: G06F-012/00

International Patent Class: G06F-12:16 ...

... G06F-11:10 ...

... G06F-11:20

Fulltext Availability:

Detailed Description

**Detailed Description**

... reserved for the new block group. The total space specified by the defined block groups includes the **parity** space **needed** to provide RAID

5 operations for all protected block groups. The blocks left over from the allocated...

...disk array will revert to transparent RAID operations, so the host must leave an adequate amount of **unallocated space** for the block pool. The amount of space necessary depends upon the access rate.

Fig. 20 shows...

15/5,K/16 (Item 10 from file: 349)  
DIALOG(R) File 349:PCT FULLTEXT  
(c) 2003 WIPO/Univentio. All rts. reserv.

00367134 \*\*Image available\*\*

**METHOD AND APPARATUS FOR IMPROVING PERFORMANCE IN A REDUNDANT ARRAY OF INDEPENDENT DISKS**

**PROCEDE ET APPAREIL PERMETTANT D'AMELIORER LA PERFORMANCE DANS UN ENSEMBLE REDONDANT DE DISQUES INDEPENDANTS**

Patent Applicant/Assignee:

BORG TECHNOLOGIES INC,

Inventor(s):

STALLMO David C,

HALL Randy K,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9707461 A1 19970227

Application: WO 96US13238 19960815 (PCT/WO US9613238)

Priority Application: US 95516293 19950817

Designated States: CA AT BE CH DE DK ES FI FR GB GR IE IT LU MC NL PT SE

Main International Patent Class: **G06F-012/00**

International Patent Class: **G06F-13:00**

Publication Language: English

Fulltext Availability:

Detailed Description

Claims

Fulltext Word Count: 13794

#### English Abstract

A RAID disk array (106, 110, 112, 114, 116) that is adaptable to host I/O traffic, wherein the RAID configuration is hidden from the host computer (102). The system dynamically determines the RAID configuration used to store host data to maximize response time performance and minimize the loss of disk space used for data protection. To maximize response time and avoid a write penalty, small write operations are mapped into RAID 1 configurations, and medium and large write operations are mapped into RAID 3 configurations. These segments are migrated into RAID 5 configurations as a background operation, to minimize the disk space lost. The system hides configuration changes necessary for the addition and/or deletion of disks to the disk array. While these changes are in progress, the disk array (106, 110, 112, 114, 116) remains on-line and all host data is available for access and modification.

#### French Abstract

Ensemble redondant (106, 110, 112, 114, 116) de disques independants RAID qui est adaptable pour accueillir un trafic E/S, dans lequel la configuration RAID est cachee par rapport a l'ordinateur central (102). Le systeme determine de maniere dynamique la configuration RAID utilisee pour stocker des donnees centrales afin de maximiser la performance de temps de reponse et pour minimiser la perte d'espace de disque utilise pour la protection des donnees. Pour maximiser le temps de reponse et eviter une penalite d'ecriture, de petites operations d'ecriture sont appliquees dans des configurations RAID 1 et des operations d'ecriture moyennes et importantes sont appliquees dans des configurations RAID 3. Ces segments sont transportes dans des configurations RAID 5 en tant qu'operation de fond, pour minimiser la perte d'espace sur disque. Ledit systeme cache les modifications de configuration necessaires pour l'ajout et/ou la suppression de disques a l'ensemble de disques. Tandis que ces



modifications sont en cours, l'ensemble (106, 110, 112, 114, 116) de disques reste en ligne et toutes les donnees centrales sont disponibles pour acces et modification.

Main International Patent Class: **G06F-012/00**  
International Patent Class: **G06F-13:00**  
Fulltext Availability:  
Detailed Description

#### Detailed Description

... reserved for the new block group. The total space specified by the defined block groups includes the **parity** space **needed** to provide RAID 5 operations for all protected block groups. The blocks left over from the allocated...

...disk array will revert to transparent RAID operations, so the host must leave an adequate amount of **unallocated space** for the block pool. The amount of space necessary depends upon the access rate.

Fig. 20 shows...

15/5,K/17 (Item 11 from file: 349)  
DIALOG(R) File 349:PCT FULLTEXT  
(c) 2003 WIPO/Univentio. All rts. reserv.

00348333 \*\*Image available\*\*

**AN INTEGRATED DEVELOPMENT PLATFORM FOR DISTRIBUTED PUBLISHING AND  
MANAGEMENT OF HYPERMEDIA OVER WIDE AREA NETWORKS  
PLATE-FORME DE DEVELOPPEMENT INTEGREE POUR LA PUBLICATION ET LA GESTION  
REPARTIES D'HYPERMEDIA SUR DES RESEAUX LONGUE PORTEE**

Patent Applicant/Assignee:

NAVISOF INC,

Inventor(s):

DOZIER Linda T,  
WILLIAMS George W V,  
LONG Dave,  
MCKEE Douglas M,  
DAVIDSON James G,  
BRADY Karen,

Patent and Priority Information (Country, Number, Date):

Patent: WO 9630846 A1 19961003

Application: WO 96US1686 19960321 (PCT/WO US9601686)

Priority Application: US 95412981 19950328

Designated States: AL AM AT AU AZ BB BG BR BY CA CH CN CZ DE DK EE ES FI GB

GE HU IS JP KE KG KP KR KZ LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL

PT RO RU SD SE SG SI SK TJ TM TR TT UA UG UZ VN KE LS MW SD SZ UG AT BE

CH DE DK ES FI FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN ML

MR NE SN TD TG

Main International Patent Class: **G06F-017/30**

Publication Language: English

Fulltext Availability:

Detailed Description

Claims

Fulltext Word Count: 177634

#### English Abstract

The present invention addresses the critical needs of publishers seeking to create and publish hypermedia content in electronic form across wide area networks ("WAN's") such as the World Wide Web. Toward this end, a client-server development platform is provided for handling the important functions of document authoring, content-based indexing and retrieval of documents, management and control of proprietary assets, and support for developing form-driven interactive services, all in a manner that is uniquely and seamlessly WAN-integrated.

#### French Abstract

Le systeme selon l'invention repond aux besoins cruciaux des editeurs desireux de creer et de publier le contenu d'hypermedia sous forme electronique dans des reseaux longue portee tels que le reseau WWW (World Wide Web). Pour ce faire, une plate-forme de developpement de serveur/client est produite pour gerer les fonctions importantes de creation de documents, indexation basee sur le contenu et d'extraction de documents, de gestion et de controle des actifs prives, et de support pour le developpement de services interactifs a base de masque, l'ensemble de maniere integree, de maniere unique et transparente aux reseaux a longue portee.

Main International Patent Class: **G06F-017/30**

Set	Items	Description
S1	425991	RESERV??? OR SAVE OR SAVING OR RETAIN??? OR (SET OR PUT) () - (ASIDE OR APART) OR PREARRANGE?
S2	22839	(UNALLOCATED OR "NOT"(1W) (ALLOCATE? OR ASSIGN? OR EARMARK? OR ALLOT??) OR UNUSED OR FREE) () (BLOCK? OR CHUNK? OR BITS OR B- BYTES OR SEGMENT? OR PIECE? OR PART? OR SPACE)
S3	8071	FILE() SYSTEM?
S4	1798316	EQUAL OR CORRESPOND? OR MATCH? OR PARALLEL? OR PARITY OR A- LIGNMENT
S5	3406	(NUMBER OR AMOUNT OR TOTAL OR SUM) () (BLOCK? OR CHUNK? OR B- ITS OR BYTES OR SEGMENT? OR PIECE? OR PART? OR SPACE)
S6	5417259	NEED? OR SIZE? OR CAPACIT? OR EXTENT OR LIMIT? OR BOUNDAR? OR THRESHOLD OR REQUIRE?
S7	23	WRITE() ANYWHERE() FILE OR WAFL
S8	15	S1 (3N) S2
S9	1	S8 AND S3
S10	1	S1 AND S2 AND S3
S11	13913	S1 AND S4 AND S6
S12	15	S11 AND S3
S13	1	S2 AND S4 AND S5 AND S6
S14	54	S7 OR S8 OR S9 OR S10 OR S12 OR S13
S15	41	S14 NOT PY>2000
S16	41	S15 NOT PD>20000818
S17	35	RD (unique items)
File	8: Ei	Compendex(R) 1970-2003/Jun W1 (c) 2003 Elsevier Eng. Info. Inc.
File	35:	Dissertation Abs Online 1861-2003/May (c) 2003 ProQuest Info&Learning
File	202:	Info. Sci. & Tech. Abs. 1966-2003/May 14 (c) Information Today, Inc
File	65:	Inside Conferences 1993-2003/Jun W2 (c) 2003 BLDSC all rts. reserv.
File	2:	INSPEC 1969-2003/Jun W1 (c) 2003 Institution of Electrical Engineers
File	233:	Internet & Personal Comp. Abs. 1981-2003/May (c) 2003 Info. Today Inc.
File	94:	JICST-EPlus 1985-2003/Jun W2 (c) 2003 Japan Science and Tech Corp(JST)
File	99:	Wilson Appl. Sci & Tech Abs 1983-2003/Apr (c) 2003 The HW Wilson Co.
File	95:	TEME-Technology & Management 1989-2003/May W4 (c) 2003 FIZ TECHNIK

17/5/1 (Item 1 from file: 8)  
DIALOG(R)File 8:EI Compendex(R)  
(c) 2003 Elsevier Eng. Info. Inc. All rts. reserv.

05384224 E.I. No: EIP99104824100

**Title: Elephant: The file system that never forgets**

Author: Santry, Douglas J.; Feeley, Michael J.; Hutchinson, Norman C.; Veitch, Alistair C.

Corporate Source: Univ of British Columbia, Vancouver, BC, Can

Conference Title: Proceedings of the 1999 7th Workshop on Hot Topics in Operating Systems (HotOS-VII)

Conference Location: Rio Rico, AZ, USA Conference Date: 19990329-19990330

Sponsor: IEEE-TCOS; AT and T; Compaq Corporation; HP Labs; et al.

E.I. Conference No.: 55660

Source: Proceedings of the Workshop on Hot Topics in Operating Systems - HOTOS 1999. p 2-7

Publication Year: 1999

CODEN: 002082

Language: English

Document Type: JA; (Journal Article) Treatment: T; (Theoretical)

Journal Announcement: 9911W3

Abstract: Modern file systems associate the deletion of a file with the release of the storage associated with that file, and file writes with the irrevocable change of file contents. We propose that this model of file system behavior is a relic of the past, when disk storage was a scarce resource. We believe that the correct model should ensure that all user actions are revocable. Deleting a file should change only the name space and file writes should overwrite no old data. The file system, not the user, should control storage allocation using a combination of user specified policies and information gleaned from file-edit histories to determine which old versions of a file to retain and for how long. This paper presents the Elephant file system, which provides users with a new contract: Elephant will automatically retain all important versions of the users files. Users name previous file versions by combining a traditional path-name with a time when the desired version of a file or directory existed. Elephant manages storage at the granularity of a file or groups of files using user-specified retention policies. This approach contrasts with checkpointing file systems such as Plan-9, AFS, and WAFL, that periodically generate efficient checkpoints of entire file systems and thus restrict retention to be guided by a single policy for all files within that file system. We also report on the Elephant prototype, which is implemented as a new Virtual File System in the FreeBSD kernel. (Author abstract) 12 Refs.

Descriptors: \*Management information systems; Magnetic disk storage; Storage allocation (computer); Computer systems programming

Identifiers: Elephant file systems

Classification Codes:

723.2 (Data Processing); 722.1 (Data Storage, Equipment & Techniques);  
723.1 (Computer Programming)  
723 (Computer Software); 722 (Computer Hardware)  
72 (COMPUTERS & DATA PROCESSING)

17/5/2 (Item 2 from file: 8)  
DIALOG(R)File 8:EI Compendex(R)  
(c) 2003 Elsevier Eng. Info. Inc. All rts. reserv.

05085921 E.I. No: EIP98084331470

**Title: Two-level signature file based on a block-oriented data model for spatial match retrieval**

Author: Chang, Chin-Chen; Lee, Chin-Feng

Corporate Source: Natl Chung Cheng Univ, Chiayi, Taiwan

Source: Journal of the Chinese Institute of Engineers, Transactions of the Chinese Institute of Engineers, Series A/Chung-kuo Kung Ch'eng Hsueh K'an v 21 n 4 Jul 1998. p 467-478

Publication Year: 1998

CODEN: JCIEEZ ISSN: 0253-3839

Language: English

Document Type: JA; (Journal Article) Treatment: G; (General Review)

Journal Announcement: 9810W1

Abstract: Signature file methods have good retrieval properties and **require** little storage overhead. Applying a signature file method to the retrieval of images from iconic image databases can reduce the number of blocks to be examined when answering a query and can thus **save** processing time. The main concern in the construction of signature files is how to minimize the average false drop rate to avoid unnecessary accesses. In this paper, we propose a two-level signature **file system** for spatial **match** retrieval based on a block-oriented data model. An algorithm is presented to determine a set of good integer solutions in the signature **size** assignment problem. In addition, our signature **file system** has taken into consideration a flexible solution to the false drop problem and other realistic factors. (Author abstract) 20 Refs.

Descriptors: \*Data structures; Database systems; Nonbibliographic retrieval systems; Data processing; Information retrieval

Identifiers: Two level signature file; Block oriented data model; Spatial **match** retrieval

Classification Codes:

723.2 (Data Processing); 723.3 (Database Systems); 903.3 (Information Retrieval & Use)

723 (Computer Software); 903 (Information Science)

72 (COMPUTERS & DATA PROCESSING); 90 (GENERAL ENGINEERING)

17/5/5 (Item 5 from file: 8)

DIALOG(R) File 8: Ei Compendex(R)

(c) 2003 Elsevier Eng. Info. Inc. All rts. reserv.

03597100 E.I. Monthly No: EI9305058959

**Title: ARIES: A transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging.**

Author: Mohan, C.; Haderle, Don; Lindsay, Bruce; Pirahesh, Hamid; Schwarz, Peter

Corporate Source: IBM Almaden Research Cent, San Jose, CA, USA

Source: ACM Transactions on Database Systems v 17 n 1 Mar 1992 p 94-162

Publication Year: 1992

CODEN: ATDSD3 ISSN: 0362-5915

Language: English

Document Type: JA; (Journal Article) Treatment: A; (Applications); G; (General Review); L; (Literature Review/Bibliography)

Journal Announcement: 9305

Abstract: In this paper we present a simple and efficient method, called ARIES (Algorithm for Recovery and Isolation Exploiting Semantics), which supports partial rollbacks of transactions, fine-granularity (e.g., record) locking and recovery using write-ahead logging (WAL). We introduce the paradigm of repeating history to redo all missing updates before performing the rollbacks of the loser transactions during restart after a system failure. ARIES uses a log sequence number in each page to correlate the state of a page with respect to logged updates of that page. All updates of a transaction are logged, including those performed during rollbacks. By appropriate chaining of the log records written during rollbacks to those written during forward progress, a bounded amount of logging is ensured during rollbacks even in the face of repeated failures during restart or of nested rollbacks. We deal with a variety of features that are very important in building and operating an industrial-strength transaction processing system. ARIES supports fuzzy checkpoints, selective and deferred restart, fuzzy image copies, media recovery, and high concurrency lock modes (e.g., increment/decrement) which exploit the semantics of the operations and **require** the ability to perform operation logging. ARIES is flexible with respect to the kinds of buffer management policies that can be implemented. It supports objects of varying length efficiently. By enabling **parallelism** during restart, page-oriented redo, and logical undo, it enhances concurrency and performance. We show why some of the System R paradigms for logging and recovery, which were based on the shadow

page technique, **need** to be changed in the context of WAL. We compare ARIES to the WAL-based recovery methods of DB2\*\*T\*\*M, IMS, and Tandem\*\*T\*\*M systems. ARIES is applicable not only to database management systems out also to persistent object-oriented languages, recoverable **file systems** and transaction-based operating systems. ARIES has been implemented, to varying degrees, in IBM's OS/2\*\*T\*\*M Extended Edition Database Manager, DB2, Workstation Data **Save** Facility/VM, Starburst and QuickSilver, and in the University of Wisconsin's EXODUS and Gamma database machine. (Author abstract) 101 Refs.

Descriptors: \*DISTRIBUTED DATABASE SYSTEMS; ALGORITHMS; INFORMATION MANAGEMENT; COMPUTER SYSTEM RECOVERY; FUZZY SETS; STORAGE ALLOCATION (COMPUTER); DATA PROCESSING

Identifiers: ALGORITHM FOR RECOVERY AND ISOLATION EXPLOITING SEMANTICS (ARIES); TRANSACTION RECOVERY METHOD; FINE GRANULARITY LOCKING; PARTIAL ROLLBACKS; WRITE AHEAD LOGGING; BUFFER MANAGEMENT

Classification Codes:

723 (Computer Software); 921 (Applied Mathematics); 722 (Computer Hardware)

72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

17/5/6 (Item 6 from file: 8)  
DIALOG(R)File 8:EI Compendex(R)  
(c) 2003 Elsevier Eng. Info. Inc. All rts. reserv.

03363096 E.I. Monthly No: EI9201002993

**Title: Design of an operating system for a scalable parallel computing engine.**

Author: Austin, Paul; Murray, Kevin; Wellings, Andy

Corporate Source: Univ of York, York, Engl

Source: Software - Practice and Experience v 21 n 10 Oct 1991 p 989-1013

Publication Year: 1991

CODEN: SPEXBL ISSN: 0038-0644

Language: English

Document Type: JA; (Journal Article) Treatment: A; (Applications); T; (Theoretical)

Journal Announcement: 9201

Abstract: There are substantial benefits to be gained from building computing systems from a number of processors working in **parallel**. One of the frequently-stated advantages of **parallel** and distributed systems is that they may be scaled to the **needs** of the user. This paper discusses some of the problems associated with designing a general-purpose operating system for a scalable **parallel** computing engine and then describes the solutions adopted in our experimental **parallel** operating system. We explain why a **parallel** computing engine composed of a collection of processors communicating through point-to-point links provides a suitable vehicle in which to realize the advantages of scaling. We then introduce a **parallel**-processing abstraction which can be used as the basis of an operating system for such a computing engine. We consider how this abstraction can be implemented and **retain** the ability to scale. As a concrete example of the ideas presented here we describe our own experimental scalable **parallel** operating-system project, concentrating on the Wisdom nucleus and the Sage **file system**. Finally, after introducing related work, we describe some of the lessons learnt from our own project. (Author abstract) 35 Refs.

Descriptors: COMPUTER SYSTEMS, DIGITAL--\* **Parallel** Processing; COMPUTER OPERATING SYSTEMS--Design; COMPUTER SYSTEMS, DIGITAL--Distributed

Identifiers: **FILE SYSTEMS**; SCALABILITY; **PARALLEL** OPERATING SYSTEMS

Classification Codes:

722 (Computer Hardware); 723 (Computer Software)

72 (COMPUTERS & DATA PROCESSING)

17/5/7 (Item 7 from file: 8)  
DIALOG(R)File 8:EI Compendex(R)  
(c) 2003 Elsevier Eng. Info. Inc. All rts. reserv.

00867847 E.I. Monthly No: EI7912093897 E.I. Yearly No: EI79022444

**Title: MATHEMATICAL MODEL FOR DISTRIBUTED FREE SPACE.**

Author: Chin, Y. H.; Yu, S. H.

Corporate Source: Cleveland State Univ, Ohio

Source: AFIPS Conference Proceedings v 48, New York, NY, Jun 4-7 1979.

Publ by AFIPS Press, Montvale, NJ, 1979 p 175-184

Publication Year: 1979

CODEN: AFPGBT ISSN: 0095-6880

Language: ENGLISH

Journal Announcement: 7912

Abstract: In commercial access methods, there are parameters provided for users to claim an amount of free space at creation. In general, a user may overestimate or underestimate the amount of distributed free space he needs. In order to determine how much distributed free space a user should claim, Y. Chin has presented a mathematical model to estimate the size of free space so that insertions do not cause the fast response to exceed the pre-set limit. The model is derived based on the worst case, namely all insertions are added into a single data storage area. As a result, that model **reserves** too much **free space**. This paper presents a new model, which reserves less distributed free storage space than Y. Chin's model, without increasing the fast response time. Characteristics of the models and consequences of experimental tests are discussed. 13 refs.

Descriptors: \*DATA PROCESSING--\*File Organization; MATHEMATICAL MODELS; DATA BASE SYSTEMS

Identifiers: DISTRIBUTED FREE SPACE

Classification Codes:

723 (Computer Software); 921 (Applied Mathematics)

72 (COMPUTERS & DATA PROCESSING); 92 (ENGINEERING MATHEMATICS)

17/5/8 (Item 8 from file: 8)

DIALOG(R)File 8:Ei Compendex(R)

(c) 2003 Elsevier Eng. Info. Inc. All rts. reserv.

00424847 E.I. Monthly No: EI7501000995 E.I. Yearly No: EI75013232

**Title: PROBLEM IN MULTIPROGRAMMED STORAGE ALLOCATION.**

Author: Ryan, Thomas A. Jr.; Coffman, Edward G. Jr.

Corporate Source: Pa State Univ, University Park

Source: IEEE Transactions on Computers v C-23 n 11 Nov 1974 p 1116-1122

Publication Year: 1974

CODEN: ITCOB4 ISSN: 0018-9340

Language: ENGLISH

Journal Announcement: 7501

Abstract: A simple mathematical model of (time-varying) program demand for main memory is developed. The model is based on the use of the immigration-death process, and is particularly suited to modeling the total demand of several programs. The goal is to study the behavior of the system under various schemes to dynamically allocating main memory among the programs. In particular, given some sort of working-set storage management the authors study what margin of **free space** should be **reserved** when programs are moved in and out of main memory, so that the frequency of overflow-underflow events is kept reasonably low, while at the same time maintaining a reasonably high degree of multiprogramming. 5 refs.

Descriptors: \*COMPUTER OPERATING SYSTEMS--\*Storage Allocation

Classification Codes:

723 (Computer Software)

72 (COMPUTERS & DATA PROCESSING)

17/5/10 (Item 2 from file: 35)

DIALOG(R)File 35:Dissertation Abs Online

(c) 2003 ProQuest Info&Learning. All rts. reserv.

01170260 ORDER NO: AAD91-22989

**DETERMINING FILE AND DIRECTORY IMPORTANCE (COGNITIVE MODELING, DECISION-MAKING)**

Author: WONG, KONG-CHENG

Degree: PH.D.  
Year: 1991  
Corporate Source/Institution: STATE UNIVERSITY OF NEW YORK AT BINGHAMTON  
(0792)  
Advisers: JOSEPH V. CORNACCHIO; ALBRECHT W. INHOFF  
Source: VOLUME 52/03-B OF DISSERTATION ABSTRACTS INTERNATIONAL.  
PAGE 1561. 143 PAGES  
Descriptors: COMPUTER SCIENCE  
Descriptor Codes: 0984

File replication is the most popular approach used to promote system reliability and file availability in a network based environment. However, all of the distributed **file systems** equipped with the functionality of file replication **require** their system users to determine how important their files are, in order to assist systems in making decisions on distributing replicas in the network (Blair et al., 1987). As such, system users are inevitably burdened with this potential responsibility. The problem can be partially alleviated if the system can take more responsibility for their system users in determining file importance. To achieve this goal, however, we **need** to better understand how system users cognitively make decisions regarding determining file importance.

We first quantitatively compare the performance of three decision-making models popularly used in juror decision-making (Pennington and Hastie, 1981) to examine how satisfactorily they model the process of determining file importance. The three models are the linear weighting model, the Bayesian model, and the Poisson model. We then propose a simple, yet powerful, decision-making model, which is called the predictor domination model, for determining file importance. The model proposed suggests that the maximum predictor values observed in the session of determining file importance may be taken as the file importance. We next examine how significantly domain-dependent information contributes to determining file importance. We demonstrate using the linear weighting model that domain-dependent information seems to contribute non-negligibly to determining file importance. Since directories are usually treated as files used to store necessary information for other files, including directories, we therefore examine how directory importance can be determined. Since a file is locatable only through its **corresponding** pathname defined by its associated tree-structured directory system, the importance of a particular directory is determined by its child files and directories having the highest importance ratings. It is also suggested that grouping those files having a higher file importance near the root will **save** not only file access time, but also the space **needed** for storing directory structures.

17/5/12 (Item 2 from file: 2)  
DIALOG(R)File 2:INSPEC  
(c) 2003 Institution of Electrical Engineers. All rts. reserv.

6732262 INSPEC Abstract Number: B2000-11-5240D-010

**Title: A hierarchical fast-multipole method for stratified media**

Author(s): Pan, Y.C.; Chew, W.C.

Author Affiliation: Dept. of Electr. & Comput. Eng., Illinois Univ., Urbana, IL, USA

Journal: Microwave and Optical Technology Letters vol.27, no.1 p. 13-17

Publisher: Wiley,

Publication Date: 5 Oct. 2000 Country of Publication: USA

CODEN: MOTLEO ISSN: 0895-2477

SICI: 0895-2477(20001005)27:1L:13:HFM;1-6

Material Identity Number: M687-2000-019

Language: English Document Type: Journal Paper (JP)

Treatment: Theoretical (T)

**Abstract:** An efficient, static, fast-multipole method (FMM)-based algorithm is presented in this paper for the evaluation of the parasitic capacitance of 3-D microstrip signal lines above stratified dielectric media. A modified tree structure is used to perform the multilevel



outgoing-to-local multipole translations. The algorithm, only marginally more expensive than the **free - space FMM**, retains its  $O(N)$  computational cost and memory use, where  $N$  is the number of conductor patches. (3 Refs)

Subfile: B

Descriptors: capacitance; inhomogeneous media; microstrip lines; numerical analysis; waveguide theory

Identifiers: hierarchical fast-multipole method; stratified media; FMM-based algorithm; parasitic capacitance; 3D microstrip signal lines; dielectric media; modified tree structure; multilevel outgoing-to-local multipole translations

Class Codes: B5240D (Waveguide and cavity theory); B1310 (Waveguides and striplines); B0290Z (Other numerical methods)

Copyright 2000, IEE

17/5/14 (Item 4 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2003 Institution of Electrical Engineers. All rts. reserv.

6473788 INSPEC Abstract Number: C2000-02-6120-039

**Title: Deciding when to forget in the Elephant file system**

Author(s): Santry, D.S.; Feeley, M.J.; Hutchinson, N.C.; Veitch, A.C.; Carton, R.W.; Ofir, J.

Author Affiliation: Dept. of Comput. Sci., British Columbia Univ., Vancouver, BC, Canada

Journal: Operating Systems Review Conference Title: Oper. Syst. Rev. (USA) vol.33, no.5 p.110-23

Publisher: ACM,

Publication Date: Dec. 1999 Country of Publication: USA

CODEN: OSRED8 ISSN: 0163-5980

SICI: 0163-5980(199912)33:5L:110:DWFE;1-Y

Material Identity Number: 0043-2000-001

Conference Title: 17th ACM Symposium on Operating Systems Principles

Conference Date: 12-15 Dec. 1999 Conference Location: Kiawah Island Resort, SC, USA

Language: English Document Type: Conference Paper (PA); Journal Paper (JP)

Treatment: Practical (P)

Abstract: Modern file systems associate the deletion of a file with the immediate release of storage, and file writes with the irrevocable change of file contents. We argue that this behavior is a relic of the past, when disk storage was a scarce resource. Today, large cheap disks make it possible for the file system to protect valuable data from accidental delete or overwrite. This paper describes the design, implementation, and performance of the Elephant file system, which automatically retains all important versions of user files. Users name previous file versions by combining a traditional pathname with a time when the desired version of a file or directory existed. Storage in Elephant is managed by the system using file-grain user-specified retention policies. This approach contrasts with checkpointing file systems such as Plan-9, AFS, and **WAFL** that periodically generate efficient checkpoints of entire file systems and thus restrict retention to be guided by a single policy for all files within that file system. Elephant is implemented as a new Virtual File System in the FreeBSD kernel. (24 Refs)

Subfile: C

Descriptors: file organisation; network operating systems; software performance evaluation; virtual storage

Identifiers: Elephant file system; file deletion; disk storage; performance; user files; directory; user-specified retention policies; checkpointing; Virtual File System; FreeBSD kernel; storage management

Class Codes: C6120 (File organisation); C6150N (Distributed systems software)

Copyright 2000, IEE

17/5/18 (Item 8 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2003 Institution of Electrical Engineers. All rts. reserv.

6080290 INSPEC Abstract Number: C9812-6160S-023

**Title: A two-level signature file based on a block-oriented data model for spatial match retrieval**

Author(s): Chin-Chen Chang; Chin-Feng Lee

Author Affiliation: Dept. of Comput. Sci. & Inf. Eng., Nat. Chung Cheng Univ., Chiayi, Taiwan

Journal: Journal of the Chinese Institute of Engineers vol.21, no.4  
p.467-78

Publisher: Chinese Inst. Eng,

Publication Date: July 1998 Country of Publication: Taiwan

CODEN: CKCKDZ ISSN: 0253-3839

SICI: 0253-3839(199807)21:4L:467:LSFB;1-P

Material Identity Number: J305-98005

Language: English Document Type: Journal Paper (JP)

Treatment: Applications (A); Theoretical (T)

Abstract: Signature file methods have good retrieval properties and **require** little storage overhead. Applying a signature file method to the retrieval of images from iconic image databases can reduce the number of blocks to be examined when answering a query and can thus **save** processing time. The main concern in the construction of signature files is how to minimize the average false drop rate to avoid unnecessary accesses. In this paper, we propose a two-level signature **file system** for spatial **match** retrieval based on a block-oriented data model. An algorithm is presented to determine a set of good integer solutions in the signature **size** assignment problem. In addition, our signature **file system** has taken into consideration a flexible solution to the false drop problem and other realistic factors. (20 Refs)

Subfile: C

Descriptors: data structures; image coding; query processing; visual databases

Identifiers: two-level signature file; block-oriented data model; spatial **match** retrieval; iconic image databases; image retrieval; signature **size** assignment problem; integer solutions; false drop problem

Class Codes: C6160S (Spatial and pictorial databases); C4250 (Database theory); C6120 (File organisation); C5260B (Computer vision and image processing techniques); C7250R (Information retrieval techniques)

Copyright 1998, IEE

17/5/21 (Item 11 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2003 Institution of Electrical Engineers. All rts. reserv.

4851706 INSPEC Abstract Number: C9502-6120-020

**Title: File system design for an NFS file server appliance**

Author(s): Hitz, D.; Lau, J.; Malcolm, M.

p.235-46

Publisher: USENIX Assoc, Berkeley, CA, USA

Publication Date: 1994 Country of Publication: USA 372 pp.

Conference Title: Proceedings of USENIX Winter 1994 Conference

Conference Date: 17-21 Jan. 1994 Conference Location: San Francisco, CA, USA

Language: English Document Type: Conference Paper (PA)

Treatment: Practical (P)

Abstract: Network Appliance Corporation recently began shipping a new kind of network server called an NFS (Network File System) file server appliance, which is a dedicated server whose sole function is to provide NFS file services. The file system requirements for an NFS appliance are different from those for a general-purpose UNIX system, both because an NFS appliance must be optimized for network file access and because an appliance must be easy to use. This paper describes **WAFL** (**Write Anywhere File** Layout), which is a file system designed specifically to work in an NFS appliance. The primary focus is on the algorithms and data structures that **WAFL** uses to implement Snapshots, which are read-only

clones of the active file system, **WAFL** uses a copy-on-write technique to minimize the disk space that Snapshots consume. This paper also describes how **WAFL** uses Snapshots to eliminate the need for file system consistency checking after an unclean shutdown. (8 Refs)

Subfile: C

Descriptors: file organisation; file servers; Unix

Identifiers: NFS file server appliance; file system design; Network Appliance Corporation; network server; Network File System; dedicated server; NFS file services; UNIX; optimization; network file access; **WAFL** ; **Write Anywhere File** Layout; data structures; Snapshots; read-only clones; active file system; copy-on-write technique; disk space minimization; file system consistency checking; unclean shutdown

Class Codes: C6120 (File organisation); C5630 (Networking equipment)

Copyright 1995, IEE

17/5/22 (Item 12 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2003 Institution of Electrical Engineers. All rts. reserv.

4518476 INSPEC Abstract Number: C9312-6150N-041

**Title: Multiprocessor file system interfaces**

Author(s): Kotz, D.

Author Affiliation: Dept. of Math. & Comput. Sci., Dartmouth Coll., Hanover, NH, USA

Conference Title: Proceedings of the Second International Conference on Parallel and Distributed Information Systems (Cat. No.93TH0493-7) p. 194-201

Publisher: IEEE Comput. Soc. Press, Los Alamitos, CA, USA

Publication Date: 1993 Country of Publication: USA xiv+272 pp.

ISBN: 0 8186 3330 1

U.S. Copyright Clearance Center Code: 0 8186 3330 1/93/\$03.00

Conference Sponsor: IEEE; ACM

Conference Date: 20-22 Jan. 1993 Conference Location: San Diego, CA, USA

Language: English Document Type: Conference Paper (PA)

Treatment: Practical (P)

Abstract: The difficulties faced when using the conventional (UNIX-like) interface in **parallel** applications are described. Ways to extend the conventional interface to provide convenient access to the file for **parallel** programs, while **retaining** the traditional interface for programs that have no need to explicitly **parallel** file access, are described. The interface includes a single naming scheme, a multiopen operation, local and global file pointers, mapped file pointers, logical records, multifiles, and logical coercion for backward compatibility. (31 Refs)

Subfile: C

Descriptors: file organisation; multiprocessing programs; **parallel** programming

Identifiers: Unix; **parallel** applications; **parallel** programs; **parallel** file access; naming scheme; multiopen operation; global file pointers; mapped file pointers; logical records; multifiles; logical coercion; backward compatibility

Class Codes: C6150N (Distributed systems); C6110P (Parallel programming); C6120 (File organisation)

17/5/25 (Item 1 from file: 233)

DIALOG(R)File 233:Internet & Personal Comp. Abs.

(c) 2003 Info. Today Inc. All rts. reserv.

00349266 94U005-007

**PC file-server power -- Network Appliance Corp.'s FAServer brings a ton of PC muscle to NFS networking**

Yager, Tom

(UNIX World's) Open Computing , May 1, 1994 , v11 n5 p83-87, 5 Page(s)

ISSN: 1072-4044

*same as  
assigned*

Company Name: Network Appliance  
Product Name: FAServer  
Languages: English  
Document Type: Hardware Review  
Grade (of Product Reviewed): B  
Hardware/Software Compatibility: IBM PC Compatible  
Geographic Location: United States

Presents a favorable review of the FAServer 400 (\$16,995) network file server from Network Appliance Corp. of Mountain View, CA (415). Says it features a single 486 processor, 16MB RAM, 2MB of battery-backed NVRAM, two 1G hard disks, RAID level 4, quick set-up and booting, excellent performance, and the **WAFL** file system; but has no serial console, no backup capability, and is limited to one telnet session at a time. Includes a photo and a product summary card. (dpm)

Descriptors: Server; Microcomputer System; Hardware Review; Network Server

Identifiers: FAServer; Network Appliance

17/5/26 (Item 2 from file: 233)

DIALOG(R) File 233:Internet & Personal Comp. Abs.  
(c) 2003 Info. Today Inc. All rts. reserv.

00315388 93IW06-373

**Dos 7 and OS/2 file systems are bitchin', but they're crusin' for a bruisin'**

Cringely, Robert X  
InfoWorld, June 28, 1993, v15 n26 p166, 1 Page(s)  
ISSN: 0199-6649  
Company Name: IBM Corp.; Microsoft  
Product Name: OS/2; MS-DOS  
Languages: English  
Document Type: Feature Articles and News  
Geographic Location: United States

NOTES FROM THE FIELD column discusses why Microsoft's DOS 7 and IBM's OS/2 are on a collision course. Says MS-DOS has always **reserved** 10 **unused bytes** for each file allocation table entry, OS/2 uses two of these bytes to store its extended file attributes. Explains that Microsoft will use these bytes in DOS 7, making OS/2 incompatible. (CR)

Descriptors: Operating Systems; OS/2; MS-DOS; Competition; Product Development

Identifiers: OS/2; MS-DOS; IBM Corp.; Microsoft

17/5/33 (Item 1 from file: 95)

DIALOG(R) File 95:TEME-Technology & Management  
(c) 2003 FIZ TECHNIK. All rts. reserv.

01116405 E97061963279

**RAMA: An easy-to-use, high-performance parallel file system**  
(Rapid Access to Massive Archive (RAMA): ein leicht zu benutzendes, ausserst leistungsfahiges System zur **parallelen** Dateiablage)  
Miller, EL; Katz, RH  
Univ. of Maryland Baltimore Country, Baltimore, USA; Univ. of California at Berkeley, USA  
Parallel Computing, v23, n4-5, pp419-446, 1997  
Document type: journal article Language: English  
Record type: Abstract  
ISSN: 0167-8191

ABSTRACT:

Modern massively **parallel** file systems provide high bandwidth file access by striping files across arrays of disks attached to a few specialised I/O nodes. However, these file systems are hard to use and difficult to integrate with workstations and tertiary storage. RAMA addresses these problems by providing a high-performance massively **parallel file system** with a simple interface. RAMA uses hashing to pseudo-randomly

distribute data to all of its disks, insuring high bandwidth regardless of access pattern and eliminating bottlenecks in file block accesses. This flexibility does not cause a large loss of performance - RAMA's simulated performance is within 10 % - 15 % of the optimum performance of a similarly- sized striped file system , and is a factor of 4 or more better than a striped file system with poorly laid out data.

DESCRIPTORS: DATA INPUT OUTPUT; **PARALLEL** PROCESSING; TIME **SAVING** ;  
PORTABILITY--SOFTWARE; ACCIDENT--CHANCE; MULTIPROCESSING SYSTEMS;  
PERFORMANCE EVALUATION; INTERNATIONAL BUSINESS MACHINES CORPORATION; MEMORY  
MANAGEMENT; OPERATING SYSTEM--COMPUTERS; EXPERIMENTAL RESULTS  
IDENTIFIERS: SKALIERBARKEIT; **paralleles File - System** ; Portabilitaet;  
Schnittstelle; RAMA

17/5/34 (Item 2 from file: 95)  
DIALOG(R)File 95:TEME-Technology & Management  
(c) 2003 FIZ TECHNIK. All rts. reserv.

01116402 E97061966279

**Evaluating approximately balanced parity -declustered data layouts for disk arrays**

(Datenorganisation gemaess der Technik ' **parity** declustering' in Arrays von Speichern: Was bringt eine nur annaehernd statt exakt ausgeglichene Lastverteilung?)

Schwabe, EJ; Sutherland, IM; Holmer, BK  
Northwestern Univ., Evanston, USA; Siemens Cupertino, USA  
Parallel Computing, v23, n4-5, pp501-523, 1997  
Document type: journal article Language: English  
Record type: Abstract  
ISSN: 0167-8191

**ABSTRACT:**

**Parity** -declustered data layouts were developed to reduce the time for on-line failure recovery in disk arrays. They generally **require** perfect balancing of reconstruction workload among the disks; this restrictive balance condition makes such data layouts difficult to construct. In this paper approximately balanced data layouts are considered, where some variation in the reconstruction workload over the disks is permitted. Such layouts are considerably easier to construct than perfectly balanced layouts. Three methods for constructing approximately balanced data layouts are considered and their performance are analyzed both theoretically and experimentally. It is concluded that on uniform workloads, approximately balanced layouts have performance nearly identical to that of perfectly balanced layouts.

DESCRIPTORS: ERROR RESILIENT SCHEME; MEMORY MANAGEMENT; DATA INPUT OUTPUT;  
INFORMATION RETRIEVAL SYSTEMS; DISTRIBUTED COMPUTING; WORST CASES;  
BREAKDOWN; MULTIPROCESSING SYSTEMS; REAL TIME METHOD; TIME **SAVING** ;  
LIKELIHOOD; MATHEMATICAL PROOF; EXPERIMENTAL RESULTS; LOAD DISTRIBUTION--  
ELECTRICAL  
IDENTIFIERS: DISK ARRAY--(DV); **PARITY** DECLUSTERING--(DV); **paralleles File - System** ; fehlertolerantes Disk Array

17/5/35 (Item 3 from file: 95)  
DIALOG(R)File 95:TEME-Technology & Management  
(c) 2003 FIZ TECHNIK. All rts. reserv.

01116398 E97061970279

**Simple randomized mergesort on parallel disks**

(Simple Randomized Mergesort: Sortieren einer Datei mit mehreren Records auf dem D-disk Model mit **paralleler** Ein-Ausgabe)

Barve, RD; Grove, EF; Vitter, JS  
Duke Univ., Durham, USA  
Parallel Computing, v23, n4-5, pp601-631, 1997  
Document type: journal article Language: English

Record type: Abstract  
ISSN: 0167-8191

ABSTRACT:

The following problem is considered: sort a file of N records on the D-disk model of **parallel** I/O in which there are two sources of **parallelism**. Records are transferred to and from disk concurrently in blocks of B contiguous records. In each I/O operation, up to one block can be transferred to or from each of the D-disks in **parallel**. A simple, efficient, randomized mergesort algorithm called SRM is proposed that uses a forecast-and-flush approach to overcome the inherent difficulties of simple merging on **parallel** disks. SRM exhibits a **limited** use of randomization and also has a useful deterministic version. Generalizing the technique of forecasting, the algorithm is able to read in, at any time, the 'right' block from any disk and using the technique of flushing, the algorithm evicts, without any I/O overhead, just the 'right' blocks from memory to make space for new ones to be read in. The disk layout of SRM is such that it enjoys perfect write **parallelism**, avoiding fundamental inefficiencies of previous mergesort algorithms. By analysis of generalized maximum occupancy problems, analytical upper bounds on SRM's expected overhead valid for arbitrary inputs are derived. The upper bound derived on expected I/O performance of SRM indicates that SRM is provably better than disk-striped mergesort (DSM) for realistic parameter values D, M, and B. Average-case simulations show further improvement on the analytical upper bound. Unlike previously proposed optimal sorting algorithms, SRM outperforms DSM even while the number of D of **parallele** disks is small.

DESCRIPTORS: DATA INPUT OUTPUT; **PARALLEL** PROCESSING; SELECTION--SORTING; ALGORITHM; MEMORY MANAGEMENT; ACCIDENT--CHANCE; PERFORMANCE EVALUATION; WORST CASES; MATHEMATICAL PROOF; COMPLEXITY; TIME **SAVING**; COMPUTERIZED SIMULATION

IDENTIFIERS: D DISK MODEL--( **FILE** **SYSTEM** ); SIMPLE RANDOMIZED MERGESORT; Sortierverfahren; Datei; D-disk model; **paralleles** I/O

Set	Items	Description
S1	1062003	RESERV??? OR SAVE OR SAVING OR RETAIN??? OR (SET OR PUT) () - (ASIDE OR APART) OR PREARRANGE?
S2	6930	(UNALLOCATED OR "NOT"(1W) (ALLOCATE? OR ASSIGN? OR EARMARK? OR ALLOT??) OR UNUSED OR FREE) () (BLOCK? OR CHUNK? OR BITS OR BYTES OR SEGMENT? OR PIECE? OR PART? OR SPACE)
S3	22396	FILE() SYSTEM?
S4	1065124	EQUAL OR CORRESPOND? OR MATCH? OR PARALLEL? OR PARITY OR ALIGNMENT
S5	2927	(NUMBER OR AMOUNT OR TOTAL OR SUM) () (BLOCK? OR CHUNK? OR BITS OR BYTES OR SEGMENT? OR PIECE? OR PART? OR SPACE)
S6	4256522	NEED? OR SIZE? OR CAPACIT? OR EXTENT OR LIMIT? OR BOUNDAR? OR THRESHOLD OR REQUIRE?
S7	81	WRITE() ANYWHERE() FILE() SYSTEM? OR WAFL
S8	5	S1 (S) S7
S9	47	S1 (3N) S2
S10	2	S9 (S) S3
S11	12	S1 (S) S2 (S) S3
S12	9	S1 (S) S4 (S) S5 (S) S6
S13	25	S8 OR S10 OR S11 OR S12
S14	22	S13 NOT PY>2000
S15	19	S14 NOT PD>20000818
S16	18	RD (unique items)

File 647: CMP Computer Fulltext 1988-2003/May W3  
(c) 2003 CMP Media, LLC

File 275: Gale Group Computer DB(TM) 1983-2003/Jun 13  
(c) 2003 The Gale Group

File 674: Computer News Fulltext 1989-2003/Jun W2  
(c) 2003 IDG Communications

File 696: DIALOG Telecom. Newsletters 1995-2003/Jun 12  
(c) 2003 The Dialog Corp.

File 98: General Sci Abs/Full-Text 1984-2003/Apr  
(c) 2003 The HW Wilson Co.

File 583: Gale Group Globalbase(TM) 1986-2002/Dec 13  
(c) 2002 The Gale Group

File 47: Gale Group Magazine DB(TM) 1959-2003/Jun 10  
(c) 2003 The Gale group

File 624: McGraw-Hill Publications 1985-2003/Jun 13  
(c) 2003 McGraw-Hill Co. Inc

File 484: Periodical Abs Plustext 1986-2003/Jun W2  
(c) 2003 ProQuest

File 141: Readers Guide 1983-2003/Apr  
(c) 2003 The HW Wilson Co

File 553: Wilson Bus. Abs. FullText 1982-2003/Apr  
(c) 2003 The HW Wilson Co

16/5,K/1 (Item 1 from file: 647)  
DIALOG(R)File 647:CMP Computer Fulltext  
(c) 2003 CMP Media, LLC. All rts. reserv.

01064388 CMP ACCESSION NUMBER: EET19950911S0084  
**NFS file server scraps Unix to ditch delays - Netapp swaps OS for  
proprietary microkernel** (mediastream)  
Loring Wirbel  
ELECTRONIC ENGINEERING TIMES, 1995, n 865, PG106  
PUBLICATION DATE: 950911  
JOURNAL CODE: EET LANGUAGE: English  
RECORD TYPE: Fulltext  
SECTION HEADING: design/Computers and Communications  
WORD COUNT: 507  
TEXT:

Mountain View, Calif. - Network Appliance Corp. (NAC) is launching the second generation of a dedicated file server for Network File System applications that NFS competitors such as Sun Microsystems and Auspex Systems might find heretical. The NetApp F330 uses a dedicated microkernel and RAID access software called File Access Software Technology (Fast), with no Unix shell present in the system.

... and MCAD departments in companies designing complex hardware.  
Srinivasan said that one software feature provided in the **WAFL** -RAID interface turned out to be of critical importance to NAC. The " Snapshot" feature provides a copy...

...use the feature to back up systems on-line, to retrieve lost files from NFS and to **save** system disk space.

The F330 has nine open PCI slots and ships with on-board support for ...

16/5,K/2 (Item 2 from file: 647)  
DIALOG(R)File 647:CMP Computer Fulltext  
(c) 2003 CMP Media, LLC. All rts. reserv.

00542886 CMP ACCESSION NUMBER: NWC19931001S1236  
**NetWare v4.0: A Rough Start for a Promising Product**  
Art Wittmann  
NETWORK COMPUTING, 1993, n 410  
PUBLICATION DATE: 931001  
JOURNAL CODE: NWC LANGUAGE: English  
RECORD TYPE: Fulltext  
SECTION HEADING: Features  
WORD COUNT: 2489  
TEXT:

There is plenty to like about NetWare v4.0, but its incompatibilities with previous products and the slow pace of add-on releases will make even the most eager NetWare proponent squeamish.

... system. You probably won't have to change hardware in existing servers to upgrade to v4.0.

**File system** improvements are evident. NetWare is now capable of dynamic read-ahead reading parts of files into memory before they actually are requested. It also supports sub-block allocation, which allows the **file system** to be chopped into larger blocks, thus **saving** memory and improving performance. NetWare achieves this by keeping information about each disk block in memory. When....

...By allowing larger disk blocks, less memory is needed, and less time must be spent searching for **free blocks**.

In general, changes like this in NetWare v4.x contribute to improved speed and stability. We noticed...

16/5,K/3 (Item 3 from file: 647)



DIALOG(R)File 647:CMP Computer Fulltext  
(c) 2003 CMP Media, LLC. All rts. reserv.

00520943 CMP ACCESSION NUMBER: NWC19921201S0178  
**SunSoft Online: DiskSuite v1.0** (Reviewed Revealed Revised)

Todd Tannenbaum  
NETWORK COMPUTING, 1992, n 314 , 24  
PUBLICATION DATE: 921201  
JOURNAL CODE: NWC LANGUAGE: English  
RECORD TYPE: Fulltext  
SECTION HEADING: Logging On  
WORD COUNT: 1268

TEXT:

Online:DiskSuite v1.0 is a software package from SunSoft that imparts to SunOS many enhanced file system capabilities, including disk mirroring, concatenating and hot spare pools. Despite minor compatibility and security limitations, we were very impressed with Online:DiskSuite, which supplies RAID-style capabilities via software at a low cost. This product has a lot to offer SunOS file systems. It improves on disk availability through disk mirroring and hot spares, it improves on performance via disk striping, and in improves on capacity through online concatenation of multiple physical drives into one file system.

... and active file system, although growfs locks out writes while doing its handiwork.

Before creating 10-GB **file systems**, be careful about some of the subtle problems of trying to grow a **file system** larger than 2 GB. For example, many application programs check the amount of **free space** on a **file system** before letting the user **save** a file. These programs are likely to report inaccurately the **free space** on any **file system** larger than 2 GB, the standard SunOS limit. Disk-Suite comes with new versions of SunOS utilities...

16/5,K/4 (Item 1 from file: 275)  
DIALOG(R)File 275:Gale Group Computer DB(TM)  
(c) 2003 The Gale Group. All rts. reserv.

02408079 SUPPLIER NUMBER: 62652933 (USE FORMAT 7 OR 9 FOR FULL TEXT)

**Special Report: Inside Windows Me Beta 3. (News Briefs)**

Finnie, Scot; Methvin, Dave

WinMag.com, NA

May 4, 2000

LANGUAGE: English RECORD TYPE: Fulltext

WORD COUNT: 7234 LINE COUNT: 00539

FILE SEGMENT: CD File 275

TEXT:

...2000 Pro, we suspect that many small businesses will look at Windows Me as a way to **save** money while keeping pace with the latest hardware technologies. The new "PC Health" features in Windows Me...free disk space drops below 200MB when SR is turned on, it will temporarily self-disable until **free space** climbs back above 200MB. You can change the amount of disk space used for SR, or manually enable/disable the feature on the Start > Settings > Control Panel > System > Performance > **File System** dialog. Nearby on the "Troubleshooting" tab, you'll also find a new option that lets you disable...

...That said, you can dig a bit inside SR's private data to see what it is **saving** each time it creates a restore point. First, make sure you go to Explorer's Tools > Folder...with a frequently clobbered file, COMDLG32.DLL. This DLL handles the chore of displaying file dialogs (open, **save**, print, etc.) for many applications. Since this file is in almost constant use, it can only be...not a huge concern. The second cool feature is shot detection. As part of the process of **saving** a video to your disk for the first time, Movie Maker scans the video and recognizes the...

16/5,K/5 (Item 2 from file: 275)  
DIALOG(R)File 275:Gale Group Computer DB(TM)  
(c) 2003 The Gale Group. All rts. reserv.

01904324 SUPPLIER NUMBER: 17964143 (USE FORMAT 7 OR 9 FOR FULL TEXT)  
**Configuring swap space. (on Solaris 2.4 and HP-UX 10.0 operating systems) (Daemons & Dragons) (Technology Tutorial) (Column)**  
McNutt, Dinah  
UNIX Review, v14, n3, p77(4)  
March, 1996  
DOCUMENT TYPE: Column ISSN: 0742-3136 LANGUAGE: English  
RECORD TYPE: Fulltext; Abstract  
WORD COUNT: 1598 LINE COUNT: 00133

ABSTRACT: Configuring swap space enables users to free up storage space and temporarily save a program running in memory. The Solaris 2.4 and HP-UX 10.0 operating systems both contain features for managing swap space. Both systems also support swapping to physical memory, a UNIX **file system** and a raw disk partition. Solaris 2.4 has no priority order and is interleaved, so all swap partitions and files are utilized equally. Using the -L option of the swap command, users can display information about each swap device and file. The -L option can also be used to determine which swap files have been configured on the system. Raw swap partitions can be added to a Solaris system by identifying an **unused partition** on a disk. Configuring swap space on HP-UX 10.0 is achieved by adding a raw partition and identifying the disk to be used as raw swap space. Once the device is identified, the swap area can be enabled using the -E option.

SPECIAL FEATURES: illustration; table; program  
DESCRIPTORS: Programming Tutorial; Operating System; Memory Manager  
SIC CODES: 7372 Prepackaged software  
TRADE NAMES: Solaris 2.4 (Operating system)--Programming; HP-UX 10.0 (Operating system)--Programming  
FILE SEGMENT: CD File 275

...ABSTRACT: both contain features for managing swap space. Both systems also support swapping to physical memory, a UNIX **file system** and a raw disk partition. Solaris 2.4 has no priority order and is interleaved, so all...

...configured on the system. Raw swap partitions can be added to a Solaris system by identifying an **unused partition** on a disk. Configuring swap space on HP-UX 10.0 is achieved by adding a raw...

16/5,K/6 (Item 3 from file: 275)  
DIALOG(R)File 275:Gale Group Computer DB(TM)  
(c) 2003 The Gale Group. All rts. reserv.

01703120 SUPPLIER NUMBER: 16240202 (USE FORMAT 7 OR 9 FOR FULL TEXT)  
**Journaling file systems: advanced network file systems offer journaling for fast recovery, greater integrity and reliability, and faster I/O. (Fault Tolerance) (Interoperability supplement to LAN Magazine )**  
Baker, Steven  
LAN Magazine, v9, n10, pS59(9)  
Oct, 1994  
ISSN: 0898-0012 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT  
WORD COUNT: 6597 LINE COUNT: 00520

ABSTRACT: Operating systems were often designed to limit disk head movement and disk access because of the slow speed of disk drives, which was often the greatest I/O bottleneck in systems. Caching information in memory saves the data to be written to disk in memory and groups it together for writing out all at once some time later. Data stored on the disk could be up to a minute behind in the event of a system failure, however, so a file system check utility is used to check and possibly

restore the file system. Journaling is used in mission-critical applications to provide for the rollback of transactions not yet completed; it is intended to have either all or none of the modifications committed to disk. Tolerant Systems was the first vendor to offer a journaling file system; its Transaction Executive (TX) operating system was bundled with its proprietary Unix hardware. IBM's AIX 3.0 for the RS/6000 included the Journaled File System (JFS). Journaling systems from Veritas, DEC, and SunSoft are described.

SPECIAL FEATURES: illustration; table

DESCRIPTORS: Backup Software; Disk/File Management Software; Fault Tolerance; Systems Software; Disaster Recovery/Prevention Software; Industry Analysis

FILE SEGMENT: CD File 275

... limits.

AdvFS is somewhat unique by conceptually treating physical disks more like logical volumes. With basic AdvFS, **file systems** (filesets in DEC jargon) are created within a domain--an entire physical disk or a partition of...

...disk. With the Advanced Utilities, this domain can become a shared pool of disks. From this domain, **file systems** can be created. Several **file systems** can be allocated from and can share disk space in a single domain (disk or disk pool). Each **file system** can have an attribute set (quota) that limits its maximum size. As long as **unused space** is available in the domain, **file systems** can be expanded merely by changing this **file system** quota. Reducing this quota will shrink a **file system** if ample file space is unused. A log device (typically 4MB) is **set aside** from each domain for any **file systems** in that domain.

With the Advanced Utilities, other features similar to Veritas VxFS are supported including file...

16/5,K/7 (Item 4 from file: 275)

DIALOG(R) File 275:Gale Group Computer DB(TM)

(c) 2003 The Gale Group. All rts. reserv.

01669455 SUPPLIER NUMBER: 15037335 (USE FORMAT 7 OR 9 FOR FULL TEXT)

Formatting SCSI disks. (the second of two parts on Small Computer System

Interface disk drives) (Daemons & Dragons) (Column) (Tutorial)

McNutt, Dinah; Pearlman, Michael

UNIX Review, v12, n3, p31(5)

March, 1994

DOCUMENT TYPE: Tutorial ISSN: 0742-3136 LANGUAGE: ENGLISH

RECORD TYPE: FULLTEXT; ABSTRACT

WORD COUNT: 1878 LINE COUNT: 00144

ABSTRACT: The second of a two-part series on Small Computer System Interface (SCSI) disk drives describes the upper levels of the disk hierarchy; the first examined the physical disk and the SCSI disk. Disks can have as many as eight partitions on most UNIX systems, each of which can be defined in any way; partitions are identified using alphabetic characters (a-h), and partition c is traditionally the whole disk. Disk partitions are used to create a file system: file-system partitions can be as large as the whole disk or as small as a single cylinder, but they always begin and end on a cylinder boundary. Information about how many partitions are on the disk and where they are located is in the disk label, which is usually located in block 0 of the disk. File systems are made up of superblocks, inodes, and data blocks; each is described, and how to determine usable disk capacity is discussed. A case study is presented of replacement of an internal SCSI drive.

SPECIAL FEATURES: illustration; chart; program

DESCRIPTORS: Disk Drives; Small Computer Systems Interface; Hard Disk Drive; Tutorial; Disk/File Management Software; Disk Space Utilization; Disk Formatting Utility; Case Study; Installing Hardware

SIC CODES: 3572 Computer storage devices  
FILE SEGMENT: CD File 275

... on the file system and increase the amount of space available for data storage.

\* m: The minimum **free - space** threshold that is **reserved** from use. This value is usually 10%. On a 1GB system, 10% is around 100MB, which is...

...value to 5%. BSD systems have a command called **tunefs** that allows you to modify the minimum **free - space** threshold after the **file system** has been created.

\* o: File systems can be optimized to either minimize the amount of time spent...

16/5,K/8 (Item 5 from file: 275)  
DIALOG(R)File 275:Gale Group Computer DB(TM)  
(c) 2003 The Gale Group. All rts. reserv.

01617276 SUPPLIER NUMBER: 14369720 (USE FORMAT 7 OR 9 FOR FULL TEXT)  
**NFS toaster. (Network Appliance Corp. FAServer Network File System server)**  
**(Hardware Review) (Tested Mettle) (includes related articles on SFS 1.0**  
**Benchmark, Network Appliance's value-added reseller program) (Evaluation)**  
UNIX Review, v11, n10, p77(6)  
Oct, 1993  
DOCUMENT TYPE: Evaluation ISSN: 0742-3136 LANGUAGE: ENGLISH  
RECORD TYPE: FULLTEXT; ABSTRACT  
WORD COUNT: 4047 LINE COUNT: 00313

ABSTRACT: Network Appliance Corp's FAServer is an 80486-based file server that redefines server hardware as a single-purpose, dedicated product. It consists of a file server, keyboard and monitor and weighs about 60 pounds; the server itself is a rugged tower containing a 50-MHz 486DX CPU and 32Mbytes of RAM. The tested configuration had one Ethernet interface module, four 1Gbyte hard disks linked in a Redundant Array of Inexpensive Disks (RAID) Level 4 arrangement, 2Mbytes of non-volatile memory, a SCSI-2 port, a VGA monitor and a standard keyboard. The FAServer's Write Anywhere File Layout (WAFL) file system combines the physical disk space of multiple disks into a single logical partition that is exported to NFS clients on the network. The machine shows attention to design detail and is simple to administer. Documentation is minimal but is very clearly written. Support policies include a one-year warranty on hardware and 90-day warranty on the bundled software. Performance is impressive, as is ease of use.

SPECIAL FEATURES: illustration; graph; table; chart  
COMPANY NAMES: Network Appliance Corp.--Products  
DESCRIPTORS: File Server; Evaluation  
SIC CODES: 3571 Electronic computers  
TRADE NAMES: Network Appliance FAServer (486-based system)--evaluation  
OPERATING PLATFORM: Intel 80486  
FILE SEGMENT: CD File 275

... mail to Network Appliance yielded the answer in a few hours. An accompanying white paper described the **WAFL** file system and RAID in more detail: One disk in the RAID level-4 array is always...

...3GB of user-available space by stating that 10% of the space on the nonparity disks is **reserved** by **WAFL** to ensure good performance in block allocation strategies in the **file system**, much like the **Fast File System** in UNIX.

Operation And Ease Of Use

The FAServer exports two directories to the network: / e t...

16/5,K/9 (Item 6 from file: 275)  
DIALOG(R)File 275:Gale Group Computer DB(TM)

(c) 2003 The Gale Group. All rts. reserv.

01310145 SUPPLIER NUMBER: 07585454 (USE FORMAT 7 OR 9 FOR FULL TEXT)

**Design goals and implementation of the new High Performance File System.**

**(includes related article on B-Trees and B+ Trees)**

Duncan, Roy

Microsoft Systems Journal, v4, n5, p1(13)

Sept, 1989

ISSN: 0889-9932

LANGUAGE: ENGLISH

RECORD TYPE: FULLTEXT; ABSTRACT

WORD COUNT: 6611

LINE COUNT: 00491

**ABSTRACT:** The High Performance File System (HPFS) enhancement to OS/2 Version 1.2 solves all the problems of the File Allocation Table (FAT) file system and is designed to meet the demands expected into the next few decades. HPFS not only serves as a way to organize data on random access block storage devices, but is also a software module that translates file-oriented requests from applications programs to device drivers. HPFS is also an example of an installable file system, which makes it possible to access several incompatible volume structures on the same OS/2 system simultaneously. Excellent throughput is achieved by the use of advanced data structures such as intelligent caching, read-ahead and write-behind. Disk space is managed more economically by the use of sectoring. HPFS also includes greatly improved fault tolerance. Applications programs need only simple modifications to make use of extended attributes and long filenames. **CAPTIONS:** Comparison of FAT and High Performance File System. (table); Overall structure of an HPFS volume. (chart); Overall structure of an Fnode. (chart)

**SPECIAL FEATURES:** illustration; table; chart

**DESCRIPTORS:** Disk/File Management Software; Enhancements; Data Structures ; Disk Space Allocation; Sectoring; OS/2

**TRADE NAMES:** OS/2 Extended Edition 1.2 (Operating system)--Product enhancement

**OPERATING PLATFORM:** OS/2

**FILE SEGMENT:** CD File 275

... of up to 16Mb each. (This maximum run size is a result of the band size and **free space** bitmap placement only; it is not an inherent limitation of the **file system**.) Reasonably small files or highly contiguous files can therefore be described completely within the Fnode (Figure 3...The primary mechanism for handling write errors is called a hotfix. When an error is detected, the **file system** takes a **free block** out of a **reserved** hotfix pool, writes the data to that block, and updates the hotfix map. (The hotfix map is...

16/5,K/10 (Item 1 from file: 674)

DIALOG(R)File 674:Computer News Fulltext

(c) 2003 IDG Communications. All rts. reserv.

077442

**Deploy software automatically**

**Today's software distribution packages aim to keep technical personnel from visiting every PC that needs an upgrade.**

Byline: JAMES GASKIN, NETWORK WORLD TEST ALLIANCE

Journal: Network World Page Number: 59

Publication Date: September 06, 1999

Word Count: 2387 Line Count: 224

**Text:**

... All the programs we tested make use of a "clean" PC that saves changes made to the **file system**, registry and desktop for distribution. WinInstall and NetInstall require a separate console for software distribution; PictureTaker and...

... pictures," and you can add, delete or modify files through a Windows Explorer-like interface. You can **save** multiple application sets that let

users get what they need as necessary or reinstall critical files that... service to pull files as well. The manual warns users to pick a server with plenty of **free space**, with good reason. WinInstall saves space for the entire directory structure of installed applications, which it copies...you create file update packages from scratch. But the point of a software distribution package is to **save** time and trouble, and tweaking application packages to death doesn't **save** either. A program as visually slick as this shouldn't cause such installation aggravation and should include...

16/5,K/18 (Item 4 from file: 484)  
DIALOG(R) File 484: Periodical Abs Plustext  
(c) 2003 ProQuest. All rts. reserv.

02729589 (USE FORMAT 7 OR 9 FOR FULLTEXT)

**Block, append: Packet writing and new file system formats**

Osborn, Kevin

CD-ROM Professional (LDP), v9 n2, p97-106

Feb 1996

ISSN: 1049-0833 JOURNAL CODE: LDP

DOCUMENT TYPE: Feature

LANGUAGE: English

RECORD TYPE: Fulltext; Abstract

WORD COUNT: 5345

LENGTH: Long (31+ col inches)

**ABSTRACT:** Packet writing is a technology that allows small chunks of data to be written, one piece at a time, to compact disc-write once media. The new packet writing technology is discussed.

Copyright Pemberton Press Inc 1996

**DESCRIPTORS:** CD-ROM; Technology

**SPECIAL FEATURES:** References Table Illustration Graph

**TEXT:**

... messages, too large a track will waste disc space and users will complain.

One alternative is to **reserve** track 1 for the primary volume descriptor (D only). The PVD is the handle to the entire ISO.9660 **file system**, since it contains the addresses of the path table and the root directory structures. There is no...

...after the last PR track, and the recording software would only have to keep track of the **free space** and stop the user before they copy files to space needed for the **file system**. Of course, with **reserving** the first track, the user must decide in advance if he or she wants to create a ...